University of Alberta

The Conjugacy Problem: Open Questions and an Application

by

Stephane R. Lemieux

A thesis sumitted to the Faculty of Graduate Studies and Research in parital fullfilment of the requirements for the degree of Doctor of Philosophy

in

Mathematics

Department of Mathematical and Statistical Sciences

Edmonton, Alberta

Fall 2004

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-95965-1
Our file Notre référence
ISBN: 0-612-95965-1

The author has granted a non-exclusive license allowing the Library and Archives Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou aturement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

To my parents Roger and Anita –

Without your endless patience, open arms, and re-opening home this latest volume of 'gibberish' would not have been possible. Sincerest thanks; I love you both.

–Stephane

Contents

1 Introduction 1

1.1 The Conjugacy and Word Problems 1

1.2 Open Problems 2

1.3 Main Results 4

2 Background 5

2.1 Novikov Groups 5

2.2 A Standard Basis for $A_{p_1p_2}$ 7

3 The Conjugacy Problem in Right-Orderable and Lattice-Orderable Groups 11

3.1 Ordered Sets 11

3.2 Ordered Groups 13

3.3 A Right-Ordering of $A_{p_1p_2}$ 17

3.3.1 G_2 is right-orderable 17

3.3.2 $A_{p_1p_2}/K$ is right-orderable 26

3.3.3 K is right-orderable 27

3.4 The Lattice-Ordered Group $L(A_{p_1p_2})$ 31

4 The Embedding Question for Torsion-Free Groups **33**

5 An application of the Conjugacy Problem **42**

5.1 The Braid Cryptosystem 42

5.2 Automata theory 46

5.3 many-variable regular languages 52

5.4 automatic groups 54

5.5 A candidate for a new cryptosystem 60

5.6 The algorithm 77

5.7 Analysis of the Algorithm 81

6 Locally Finite-Indicable Groups **84**

6.1 Groups Without Normal in $\mathfrak{F}$ 85

6.2 Two-generator Groups 87

6.2.1 Embedding in a Two-generator Group 88

6.2.2 Two-generator Groups in $L(\mathfrak{F})\backslash N(\mathfrak{F})$ 89

Chapter 1

Introduction

1.1 The Conjugacy and Word Problems

The conjugacy problem for a group G is the problem of determining, given $x, y \in G$, whether or not there exists an element $z \in G$ such that $z^{-1}xz = y$. If there is an algorithm which, for any $x, y \in G$, always terminates in a finite number of steps with a definite yes or no, answer then the conjugacy problem for G is said to be solvable. Otherwise it is said to be unsolvable. The problem of producing such a z, given that x and y are conjugate in G is called the generalized conjugacy problem. The word problem for G is similar and consists of determining , given $x, y \in G$, whether or not $x = y$ in G. An equivalent characterization of the word problem is the problem of determining, given $v \in G$, whether or not v is equal to the identity element in G, which will always be denoted e in this thesis. The equivalence of the

previous two problems is obvious if one considers that $x = y$ if and only if $y^{-1}x = e$.

The word and conjugacy problems are not equivalent but they are closely linked because $v = e$ if and only if $\exists x \in G$ such that $x^{-1}vx = e$ if and only if $\forall x \in G$, $x^{-1}vx = e$. This means a solution to the conjugacy problem implies a solution to the word problem.

Another related problem which is even stronger than the word problem is called the power problem. This is the problem of determining algorithmically, given two elements x, and y of a group G, whether or not there exists an integer k such that $x^k = y$.

1.2 Open Problems

This thesis improves results on three open problems in group theory and discloses a new cryptosystem which uses automatic and word hyperbolic groups in place of braid groups, to improve on the speed and security of the braid cryptosystem. The braid cryptosystem was shown in [15] to be insecure. The open problems are as follows.

1. Does there exist a finitely presented group which has solvable word problem, unsolvable conjugacy problem and is right-orderable?

2. Can every torsion-free group with solvable word problem be embedded in a group with solvable conjugacy problem?

3. Is the class of locally finite-indicable groups equal to the class of groups which have a normal system with finite factors?

Although this thesis does not deal exclusively with the conjugacy problem the title for the thesis is still appropriate because the first two problems deal with the conjugacy problem, and the unsolvability of the conjugacy problem for the group used in the cryptosystem is essential to its security. The relevant chapters require a majority of the time and effort and page are the longest. The last problem is distinctive and is given its own chapter, in which we define the terminology and provide previous results that are distinctly pertinent here.

An affirmative answer to the first problem would be an important step toward solving the problem posed by A.M.W. Glass in [12] as to the existence of a finitely-presented lattice-ordered group with solvable word problem and unsolvable conjugacy problem. It is listed as one of the primary open problems in the study of lattice-ordered groups. Question two, posed by D. J. Collins, is well known among group theorists and remains open to this day. A lesser result, showing that every torsion-free group with solvable power-problem can be embedded in a group with solvable conjugacy problem is cited in the paper [25]. However, it was proved as a corollary to a more complicated result, the proof of which is very complicated and draws on a lot of previous work of the authors of [25]. We therefore give a much simpler and more self-contained proof of the result directly.

Question three was suggested to the author by Akbar Rhemtulla and is answered conclusively in chapter six.

1.3 Main Results

This section of the text details the main results which answer the above questions. Subsequent chapters will be devoted in turn to proving the following results.

In Chapter 3, we prove the following result in response to question one.

Theorem 1 *There exists a finitely presented group G, with solvable word problem and unsolvable conjugacy problem, that is right-orderable.*

In Chapter 4 we give a more direct, simple, and self-contained proof to the following already known result.

Theorem 2 *Every torsion-free group with solvable power problem can be embedded in a torsion-free group with solvable conjugacy problem.*

In Chapter 5, we exhibit groups which have all of the desirable characteristics of the braid group as well as extra characteristics which allow for faster key exchange and increased security in a public key cryptosystem.

As stated earlier, Chapter 6 is a self-contained paper in which we construct a family of finitely generated groups that are locally finite-indicable but do not have a normal system with finite factors.

Chapter 2

Background

It will be assumed that the reader is familiar with the fundamental concepts of group theory, but when the necessary concepts for dealing with each open question are not fundamental to group theory, these will be examined in the appropriate chapter immediately before proving the corresponding main result. The exception to this convention is the following section on Novikov Groups, which are used in the next chapter.

2.1 Novikov Groups

It is known that for any recursively enumerable degree of unsolvability there is a finitely presented semigroup of the form $S = \langle a_j, A_i = B_i \mid 1 \leq i \leq \lambda, 1 \leq j \leq n \rangle$ whose word problem is of that degree. It was shown by Bokut that the degree of unsolvability of the word problem for S is equal to the

degree of unsolvability of the conjugacy problem for $A_{p_1p_2}$, when $A_{p_1p_2}$ is defined using S as follows.

The group $A_{p_1p_2}$ is most easily dealt with when defined via an ascending sequence of four groups as follows:

- $G_0 = \langle \Sigma_0; \Phi_0 \rangle$
- $G_1 = \langle \Sigma_0 \cup \Sigma_1; \Phi_0 \cup \Phi_1 \rangle$
- $G_2 = \langle \Sigma_0 \cup \Sigma_1 \cup \Sigma_2; \Phi_0 \cup \Phi_1 \cup \Phi_2 \rangle$
- $A_{p_1p_2} = G_3 = \langle \cup_{i=0}^{3} \Sigma_i; \cup_{i=0}^{3} \Phi_i \rangle$

Where

- $\Sigma_0 = \{q_i, t_i, q_i^+, t_i^+, 1 \leq i \leq \lambda\}$
- $\Sigma_1 = \{a_j, a_j^+, 1 \leq j \leq n\}$
- $\Sigma_2 = \{l_i, l_i^+, 1 \leq i \leq \lambda\}$
- $\Sigma_3 = \{p_1, p_2\}$
- $\Phi_0 = \emptyset$
- $\Phi_1 = \{q_i a_j = a_j q_i^2,\ t_i^2 a_j = a_j t_i,\ a_j^+ q_i^+ = (q_i^+)^2 a_j^+,\ t_i^+ a_j^+ = a_j^+ (t_i^+)^2\}$
- $\Phi_2 = \{l_i a_j = a_j l_i,\ l_i^+ a_j^+ = a_j^+ l_i^+\}$
- $\Phi_3 = \{(A_i^+)^{-1} q_i^+ l_i^+ p_1 = p_1 A_i q_i^{-1} l_i^{-1},\ (t_i^+)^{-1} p_1 = p_1 t_i,\ B_i^{-1} t_i l_i p_2 = p_2 B_i^+ (t_i^+)^{-1} (l_i^+)^{-1},\ q_i^{-1} p_2 = p_2 q_i^+\}$

with A_i's and B_i's distinct words in $\langle a_j, 1 \leq j \leq n \rangle$.

2.2 A Standard Basis for $A_{p_1p_2}$

A group G has a standard basis if there exists a subset L of words in the generators of G, and a bijection between L and G, such that each element of the basis is equivalent to one and only one group element. Thus if we assume the axiom of choice, then technically every group has a standard basis. However, giving an explicit finite presentation of the basis is usually not possible because of the dependence on the axiom of choice. Therefore we reserve the term standard basis for those groups that have a finite, or at least recursive, presentation.

Even in this stricter sense of the term, $A_{p_1p_2}$ has a standard basis which is defined in terms of the ascending sequence of groups $G_0 \subset G_1 \subset G_2 \subset G_3 = A_{p_1p_2}$ as follows:

Each of the sets C_0, C_1, C_2, and C_3 is a standard basis for G_0, G_1, G_2, and G_3 respectively.

C_0 consists of all irreducible group words of the alphabet Σ_0. A word is said to be irreducible if it does not contain subwords of the form xx^{-1} or $x^{-1}x$.

C_1 consists of all words of the from

$$(**)\ w = u_1 x_1^{\epsilon_1} u_2 x_2^{\epsilon_2} \ldots u_k x_k^{\epsilon_k} u_{k+1},$$

where $k \geq 0$, $x_i \in \Sigma_1$, $\epsilon_i = \pm 1$, $u_i \in C_0$ and w is irreducible and does not contain the subwords:

1. $q_i^{\epsilon} a_j$, $q_i^{-2} a_j^{-1}$, $q_i a_j^{-1}$, $t_i^2 a_j$, $t_i^{-1} a_j$, $t_i^{\epsilon} a_j^{-1}$

2. $(q_i^+)^2 a_j^+,\ (q_i^+)^{-1} a_j^+,\ (q_i^+)^\epsilon (a_j^+)^{-1},\ (t_i^+)^\epsilon a_j^+,\ t_i^+ (a_j^+)^{-1},\ (t_i^+)^{-2}(a_j^+)^{-1}$

where $\epsilon = \pm 1,\ 1 \le i \le \lambda,\ 1 \le j \le n$.

C_2 consists of irreducible words of the form of $(**)$ in which $k \ge 0,\ u_i \in C_1,\ x_i \in \Sigma_2,\ \epsilon_i = \pm 1$, and which do not contain the following subwords:

3. $a_j V(q_s^2, t_s) l_i^\epsilon,\ a_j V(q_s, t_s^2) l_i^\epsilon$

4. $a_j^+ V(a_s^+, (t_s^+)^2)(l_i)^\epsilon,\ (a_j^+)^{-1} V((q_s^+)^2, t_s^+)(l_i^+)^\epsilon$

where $\epsilon = \pm 1,\ 1 \le j \le n,\ 1 \le i \le \lambda$ and $V(x, y)$ are irreducible words of G_0 in x and y.

C_3 consists of irreducible words of the form $(**)$ in which $k \ge 0,\ u_i \in C_2,\ x_i \in \Sigma_3,\ \epsilon_i = \pm 1$, and which do not contain the following subwords:

5. $(t_i^+)^\epsilon p_1,\ t_i^\epsilon p_1^{-1},\ q_i^\epsilon p_2,\ (q_i^+)^\epsilon p_2^{-1}$

6. $l_i^+ V(a_j^+) W(t_j^+) p_1,\ l_i^{-1} V(a_j) W(t_j) p_1^{-1}$

7. $l_i V(a_j) W(q_j) p_2,\ (l_i^+)^{-1} V(a_j^+) W(q_j^+) p_2^{-1}$,

8. $(l_i^+)^{-1} V(a_j^+) C((q_i^+)^{-1} A_i^+) W(t_j^+) p_1,\ l_i V(a_j) C(q_i A_i^{-1}) W(t_j) p_1^{-1}$,

9. $l_i^{-1} V(a_j) C(t_i^{-1} B_i) W(q_j) p_2,\ l_i^+ V(a_j^+) C(t_i^+ (B_i^+)^{-1}) W(q_j^+) p_2^{-1}$

where V and W are reduced words and $C(U)$ denotes a canonical word equal to U.

The reader has probably noticed that $A_{p_1 p_2}$ does not, in the strictest sense refer to a single group but rather a family of groups because distinct

choices for the semigroup S produce distinct examples of $A_{p_1p_2}$. This family of groups was first discovered by Novikov in [24] and shown by Bokut in [4] to have a standard basis, so we refer to the family as Novikov groups. Thus, in this text, by $A_{p_1p_2}$ we refer to an example of a Novikov group. Any theorem proved for $A_{p_1p_2}$ will hold for any Novikov group.

It was Bokut in [5] who proved that for any recursively enumerable degree d of unsolvability, there is a Novikov group whose conjugacy problem has degree d. He also proved that the word problem for $A_{p_1p_2}$ is solvable and the conjugacy problem for each of G_0, G_1, and G_2 is solvable.

We have adopted Bokut's notation when dealing with $A_{p_1p_2}$ (except that our t_i's are actually r_i's in his paper but they look like tao's). Also the symbol G_2', in this paper as well as his, refers to the group $\langle a_j, q_i, t_i, l_i \mid 1 \leq j \leq \lambda,\ 1 \leq i \leq n \rangle$. Thus $G_2' \neq [G_2, G_2]$. However for any other group G, H or K, in this text, we keep with convention, i.e. $G' = [G, G]$ the commutator subgroup of G.

We end this section with a statement of Britton's Lemma with a preliminary, explanatory excerpt from [3].

Let $\overline{G} = \langle \Sigma; \Phi \rangle$ be a group with generators Σ and relations Φ. The lemma was proved by Britton as a tool, useful in dealing with groups like $A_{p_1p_2}$. In fact we shall rely on it several times in Chapter 3 when proving that $A_{p_1p_2}$ is right-orderable.

Let $\overline{G} = \langle \Sigma; \Phi \rangle$ be a group with generators Σ and relations Φ. The group

$$G = \langle \Sigma,\ \mathcal{B};\ \Phi,\ A_i p_{m_i} = p_{n_i} B_i,\ i \in I \rangle$$

where $\Sigma \bigcap \mathcal{B}$ is empty, $p_{m_i}, p_{n_i} \in \mathcal{B}$ and A_i, B_i are Σ-words in the group with stable letters $\mathcal{B}$ and base group $\overline{G}$.

Lemma 1 (Britton's lemma [3]) *Let $\mathcal{B}$ be a regular system of stable letters of the group G, with base group $\overline{G}$ and let W be a $(\Sigma \bigcup \mathcal{B})$-word. If $W = e$ in G then either W is a Σ-word and $W = e$ in $\overline{G}$ or W contains a subword of the form $p_n^{-\epsilon} U p_m^{\epsilon}$ where U is a Σ-word and for some $U = \mathcal{U}_{p_m^{\epsilon} p_n^{\epsilon}}$.*

By a σ-word, where σ is an alphabet, we mean a group word constructed from this alphabet. A system of stable letters $\mathcal{B}$ of the group G is a subset of the defining alphabet of G such that no relation of G decreases the number of occurences of $\mathcal{B}$-letters in any word in G, except the trivial relations where stable letters are juxtaposed with their inverses. For example p_1 and p_2 are the stable letters of $A_{p_1 p_2}$, the l_i's are the stable letters of G_2, and the a_j's are the stable letters of G_1. A system of stable letters is regular if for every relation $A_i p_{m_i} = p_{n_i} B_i$, $B_i = e$ if and only if $A_i = e$. Finally, a word $U = \mathcal{U}_{p_m^{\epsilon} p_n^{\epsilon}}$ is simply a product of A_i's and/or B_i's such that $p_n^{-\epsilon} U p_m^{\epsilon} = p_n^{-\epsilon} p_n^{\epsilon} U'$ or $p_n^{-\epsilon} U p_m^{\epsilon} = U'' p_m^{-\epsilon} p_m^{\epsilon}$, for some U' or U''.

Chapter 3

The Conjugacy Problem in Right-Orderable and Lattice-Orderable Groups

3.1 Ordered Sets

The following definitions and results on partially-ordered sets, lattice-ordered groups and right-ordered groups are reproduced from [18] and [19].

A non-empty set M is called partially-ordered if it is equipped with a binary relation $\leq$ on M satisfying the axioms:

1. $\forall x \in M$, $x \leq x$

2. $\forall x, y \in M$, if $x \leq y$ and $y \leq x$ then $x = y$

3. $\forall x, y, z \in M$, if $x \leq y$ and $y \leq z$ then $x \leq z$.

This binary relation $\leq$ is called the partial-order on the set M. If $x \leq y$ or $y \leq x$, then x and y are said to be comparable, otherwise they are incomparable; $x < y$ means $x \leq y$ and $x \neq y$.

A partially-ordered set M is totally-ordered or linearly-ordered if every two elements of M are comparable. A totally-ordered set M is well-ordered if every non-empty subset of M has a least element. It is conventional in ordered-group theory to suppose that Zermelo's theorem is true (it is equivalent to the axiom of choice), i.e., any arbitrary set can be well-ordered.

Let x, y be elements of the partially-ordered set M. If $\exists u \in M$, such that $x \leq u$ and $y \leq u$, then u is an upper bound for x and y. Lower bound is defined analogously. If there exists an upper bound z for x and y, such that $z \leq u$ for every upper bound u of x and y, then z is called the least upper bound or join of x and y and is denoted $x \vee y$. The greatest lower bound or meet is defined analogously and is denoted $x \wedge y$. A partially ordered set M for which $x \vee y$ and $x \wedge y$ exist $\forall x, y \in M$ is called a lattice-ordered set, or simply a lattice. Note that every totally-ordered set is also a lattice-ordered set but the converse is not true.

Any lattice can be characterized by the following identities:

1. $x \vee x = x, \quad x \wedge x = x$

2. $x \vee y = y \vee x, \quad x \wedge y = y \wedge x$

3. $(x \vee y) \vee z = x \vee (y \vee z), \quad (x \wedge y) \wedge z = x \wedge (y \wedge z)$

4. $(x \vee y) \wedge x = x, \quad (x \wedge y) \vee x = x,$

as demonstrated in the following theorem.

Theorem 3 *[19] For any lattice L, the identities 1 to 4 are valid in L. Conversely, let L be an algebraic system of signature $\{\vee, \wedge\}$ such that identities 1 to 4 are valid in L. Then L is a lattice under the partial order defined by the rule: $x \leq y$ if and only if $x \vee y = y$.*

The lattice L is called distributive if $\forall x, y, z \in L$ the following are valid:

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$$

and

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z).$$

There is a weaker notion of a modular lattice but it is not necessary for this report since every distributive lattice is modular. Note that any totally-ordered set is a distributive lattice. The class of distributive lattices is closed under cardinal products, homomorphisms and sublattices.

Theorem 4 *[19] Any distributive lattice is isomorphic to some sublattice of the lattice $S(M)$ of subsets for some set M.*

3.2 Ordered Groups

A partially-ordered group is a non-empty set G with binary operation $\cdot$ and a binary operation $\leq$ such that $\{G; \cdot\}$ is a group and $\{G; \leq\}$ is a partially-ordered set and the following axioms are fulfilled:

1. $\forall x, y, z \in G$, $x \leq y$ implies $xz \leq yz$

2. $\forall x, y, z \in G$, $x \leq y$ implies $zx \leq zy$

If the order on G is a lattice, then G is called a lattice-ordered group (l-group). If the order is a total-order, then the G is a totally-ordered or just an ordered group (o-group).

The definition of a partially-ordered group stipulates that the order must be invariant under multiplication from both sides. However, there is the notion of a group which is invariant under multiplication only on the right-hand-side. If the first of the axioms above hold then G is called partially right-ordered. If the order is total then G is called a right-ordered group. Note that if a group is partially right-ordered then multiplication on the left by inverses gives a partial left-order so every partially right-orderable group is also partially left-orderable but not necessarily simultaneously, i.e., with respect to the same order.

Let G be a partially right-ordered group. An element $x \in G$ is called positive (strictly positive) if $x \geq e$ ($x > e$). It is negative (strictly negative) if $x \leq e$ ($x < e$). The set of positive elements of a partially right-ordered group G is called the positive cone.

Theorem 5 *[18] For a partially right-ordered group G with a positive cone P, the following relations hold:*

$$P \cdot P \subseteq P$$

$$P \cap P^{-1} = \{e\}$$

If G is a right-ordered group, then

$$G = P \cup P^{-1}$$

Conversely, if in a group G there is a subset P satisfying the first two relations then it is possible to introduce a partial-order $\leq$ on G such that $\{G; \cdot; \leq\}$ is a partially right-ordered group with the positive cone P. If P also satisfies the third relation, then G is a right-ordered group.

Theorem 6 *[18] A partially right-ordered group G with positive cone P is a partially-ordered group if and only if P satisfies the first two relations of the previous theorem and also satisfies:*

$$x^{-1} \cdot P \cdot x \subseteq P, \ \forall x \in G$$

Theorem 7 *[18] A partially-ordered group G with positive cone P is a lattice-ordered group if and only if it is directed and P is a lattice with respect to the induced order.*

Theorem 8 *[18] The free product $G*$ of right-ordered $\{G_\alpha \mid \alpha \in I\}$ is a right-orderable group, and for every group G_α its right-order can be extended to a right-order on the group $G*$.*

Theorem 9 (corollary to Kurósh Subgroup Theorem [22]) *Let G be a free product of A, B, C with amalgamations from the factor A, i.e., all*

defining relations either involve one type of generator, or have the form $U(a_\nu) = V(b_\mu$ *or* $U(a_\nu = W(c_\zeta)$. *Then any subgroup* H *of* G, *whose intersection with the conjugates of* A, B, *and* C *is* e, *must be a free group.*

Theorem 10 *[18] A group* G *is fully-orderable (right-orderable) if and only if every finitely generated subgroup is fully-orderable (right-orderable).*

Lattice-ordered, right-ordered, and totally-ordered groups share the property that they must all be torsion free, i.e. $x \neq e \Rightarrow x^n \neq e, \forall n$. It is also true that, for any element of such a group, $x \geq e \Rightarrow x^{-1} \leq e$ and $x \geq e \Rightarrow x^n \geq e, \forall n > 0$

However, there are many fundamental group theoretic properties that they do not share as the following results indicate.

Theorem 11 *[19] If* G *is an ordered group or a right-ordered group and* H *is any subgroup of* G *then* H *is ordered or right-ordered respectively. If* G *is a lattice-ordered group then* H *need not be a lattice-ordered group.*

Theorem 12 *[18] (Levi) Let* N *be a normal subgroup of a group* G, P_N *be a partial right-order on the group* N, *and* $\overline{P}$ *be a partial right-order on the quotient group* $\overline{G} = G/N$. *Then there is a partial right-order* P *on the group* G *such that* (G, P) *is the lexicographic extension of* (N, P_N) *by* $(\overline{G}, \overline{P})$. *If the groups* (N, P_N) *and* $(\overline{G}, \overline{P})$ *are partially-ordered and* $g^{-1}P_N g = P_n$ *for any* $g \in G$, *then the group* (G, P) *is also partially-ordered if* $\overline{P}$ *is a partial-order on* $\overline{G}$.

According to this theorem, for a lattice-ordered normal subgroup to have a lexicographic extension, it must be invariant under conjugation by the extension group.

Theorem 13 (unique extraction of roots [19]) *If G is a totally-ordered group then $\forall x, y \in G,\ x^n = y^n \Rightarrow x = y$. If G is a lattice-ordered group then $\forall x, y \in G,\ x^n = y^n \Rightarrow \exists z \in G$ such that $z^{-1}xz = y$*

3.3 A Right-Ordering of $A_{p_1p_2}$

In this section we prove theorem 1 of the main results, i.e. the existence of a finitely presented group, which admits a right-ordering and has solvable word problem and unsolvable conjugacy problem. We do so by proving the following theorem.

Theorem 14 *The group $A_{p_1p_2}$ is right-orderable.*

We prove the above result by defining the normal series $A_{p_1p_2} \lhd HK \lhd K$ and constructing right-orders on $A_{p_1p_2}/HK$, HK/K and K separately. Theorem 12 then implies that $A_{p_1p_2}$ is right-orderable. First, however, we need the following result.

3.3.1 G_2 is right-orderable

Lemma 2 *The subgroup G_2 of $A_{p_1p_2}$ is right-orderable.*

To show that G_2 is right-orderable, it is enough to construct a right-order on G_2' because G_2 is the free product of anti-isomorphic subgroups G_2' and G_2^+. We begin by labeling certain subgroups of G_2' for easier reference.

$$A = \langle a_1, ..., a_n \rangle$$

$$Q = \langle q_1, ..., q_\lambda \rangle$$

$$L = \langle l_1, ..., l_\lambda \rangle$$

$$T = \langle t_1, ..., t_\lambda \rangle$$

Let $B = (Q * T * L)^A = \langle u^{-1}vu \mid u \in A, v \in Q * T * L \rangle$. Then by definition $A \leq N_{G_2'}(B)$ so B is normal in G_2' with $G_2' = AB$. Furthermore, $G_2'/B = AB/B \cong A \cong F_n$ so G_2'/B is right-orderable. As usual F_n denotes a free-group of rank n. Thus to show G_2' is right-orderable, by Theorem 12 it is sufficient to show that B is right orderable.

Recall the relations of G_2' are

$$a_j^{-1} q_i a_j = q_i^2, \ for \ 1 \leq i \leq \lambda, \ 1 \leq j \leq n,$$

$$a_j t_i a_j^{-1} = t_i^2, \ for \ 1 \leq i \leq \lambda, \ 1 \leq j \leq n, \ and$$

$$a_j^{-1} l_i a_j = l_i \ for \ 1 \leq i \leq \lambda, \ 1 \leq j \leq n.$$

In light of these relations we can think of elements of B of the form $u^{-1}xu$ where $u \in A$ and $x \in \{q_i^j, t_i^j\}$ as k-th roots of $\langle x \rangle$ because we will show that for each $u^{-1}xu$ there exists a smallest positive integer k such that $(u^{-1}xu)^k \in \langle x \rangle$.

It is obvious, as there are no non-trivial relations which hold in $\langle Q, T, L\rangle$, that $B = (\langle q_1\rangle * \langle q_2\rangle * ... * \langle q_\lambda\rangle * \langle t_1\rangle * ... * \langle t_\lambda\rangle * \langle l_1\rangle * ... * \langle l_\lambda\rangle)^A$. We now show, using Britton's Lemma, that in fact

$$B = \langle q_1\rangle^A * \langle q_2\rangle^A * ... * \langle q_\lambda\rangle^A * \langle t_1\rangle^A * ... * \langle t_\lambda\rangle^A * \langle l_1\rangle * ... * \langle l_\lambda\rangle.$$

Recall that we used $\{l_1, ..., l_\lambda\}$ as the stable letters of G_2' because $l_i^{-1} a_j l_i = a_j$ and that we used $\{a_1, ..., a_n\}$ as the stable letters of G_1' because $a_j^{-1} q_i a_j = q_i^2$ and $a_j t_i a_j^{-1} = t_i^2$. However we could also view $\{a_1, ..., a_n\}$ as the stable letters of G_2' because $a_j^{-1} l_i a_j = l_i$, so long as we realize that the base group would then be $\langle Q, T, L\rangle$ instead of G_1'.

With this new set of stable letters, suppose that $R = e$ is a relation that holds in B. Then R is a word in the generators (and their inverses) of B that is equal to e in B and hence in G_2'. Therefore, by Britton's Lemma, either R is a word in the generators (and their inverses) of $\langle Q, T, L\rangle$, or there exists a pinch of the form $a_j^{-\epsilon} U a_j^{\epsilon}$ where U is a word in the generators (and their inverses) of $\langle Q, T, L\rangle$ and $\epsilon = \pm 1$ and $a_j^{-\epsilon} U a_j^{\epsilon} = U' a_j^{-\epsilon} a_j^{\epsilon}$. Therefore U is generated by:

- $\{q_i^{\pm 1}, l_i^{\pm 1}, t_i^{\pm 2} \mid 1 \leq i \leq \lambda\}$ if $\epsilon = 1$
- $\{q_i^{\pm 2}, l_i^{\pm 1}, t_i^{\pm 1} \mid 1 \leq i \leq \lambda\}$ if $\epsilon = -1$.

Since $R = e$, we can perform as many pinches of the above form as necessary until we arrive at $R = R_2$ where R_2 is a word in $\{q_i^{\pm 1}, l_i^{\pm 1}, t_i^{\pm 1}\}$ and $R_2 = e$. But $\{q_i^{\pm 1}, l_i^{\pm 1}, t_i^{\pm 1}\}$ generates a free group so R_2 freely reduces to

the identity, i.e., $R = v_1 v_2 ... v_\alpha$ such that for each v_j there is a fixed x_i from $\{q_i, l_i, t_i\}$ such that v_j is a word in powers of x_i, with the sum of the powers being 0. This proves that B is the free product we claimed because R must be the word $v_1' v_2' ... v_\alpha'$ where $v_j' = v_j$ in G_2' and so $v_j' \in (\langle x_i \rangle)^A$ where v_j is a word in $\{x_i^{\pm 1}\}$.

Therefore, Theorem 8 implies that, to show B is right-orderable, we need only show that $\langle q_1 \rangle^A = \langle u^{-1} q_1 u \mid u \in A \rangle$ is right-orderable because B is a free product of groups isomorphic to $\langle q_1 \rangle^A$. Theorem ..... further reduces the task to showing that all finitely generated subgroups of $\langle q_1 \rangle^A$ are right-orderable. For ease of notation, as it does not matter which q_i we demonstrate on, let $q_1 = q$.

We begin by showing that every subgroup of $\langle q \rangle^A$ generated by two elements is right-orderable. Fix $u_1, u_2 \in A$ and consider the group $\langle u_1^{-1} q u_1, u_2^{-1} q u_2 \rangle$. In actuality the most general form of a subgroup of $\langle q \rangle^A$ generated by two elements would be $\langle u_1^{-1} q^{k_1} u_1, u_2^{-1} q^{k_2} u_2 \rangle$ where k_1 and k_2 are integers but $\langle u_1^{-1} q^{k_1} u_1, u_2^{-1} q^{k_2} u_2 \rangle$ is a subgroup of $\langle u_1^{-1} q u_1, u_2^{-1} q u_2 \rangle$, so right-orderability of the latter implies right orderability of the former.

Lemma 3 *There exist integers n_1, n_2 such that $(u_1^{-1} q u_1)^{n_1} \in \langle q \rangle$, and $(u_2^{-1} q u_2)^{n_2} \in \langle q \rangle$. We may assume that n_1, n_2 have the smallest magnitude possible.*

Proof: Since $u_1 = a_{i_1}^{\alpha_1} a_{i_2}^{\alpha_2} ... a_{i_k}^{\alpha_k}$, such that α_i are integers and each $a_{i_j} \in \{a_1, ..., a_n\}$, we set $d_1 = min\{\alpha_1, \alpha_1 + \alpha_2, ..., \Sigma_{i=1}^{k} \alpha_i\}$.

First we show that if $d_1 \geq 0$ then $u_1^{-1}qu_1 \in \langle q \rangle$ and $n_1 = 0$. Proceeding by induction, $\alpha_1 \geq 0$ so $a_{i_1}^{-\alpha_1} q a_{i_1}^{\alpha_1} = q^{2^{\alpha_1}} \in \langle q \rangle$. By the inductive assumption $(a_{i_1}^{\alpha_1} a_{i_2}^{\alpha_2} ... a_{i_j}^{\alpha_j})^{-1} q a_{i_1}^{\alpha_1} a_{i_2}^{\alpha_2} ... a_{i_j}^{\alpha_j} = q^{2^{\Sigma_{i=1}^{j} \alpha_i}} \in \langle q \rangle$. Thus $a_{j+1}^{-\alpha_{j+1}} q^{2^{\Sigma_{i=1}^{j} \alpha_i}} a_{j+1}^{\alpha_{j+1}} = q^{2^{\Sigma_{i=1}^{j+1} \alpha_i}}$ which is in $\langle q \rangle$ since $\Sigma_{i=1}^{j+1} \alpha_i \geq 0$.

Otherwise if $d_1 < 0$ then let $n_1 = 2^{-d_1} > 0$. Then $(u_1^{-1}qu_1)^{n_1} = u_1^{-1} q^{2^{-d_1}} u_1 = q^{2^{-d_1 + \Sigma_{i=1}^{k} \alpha_i}} \in Q$. We can find n_2 in the same manner so the proof is complete.

To illustrate the method we use the following example. Let

$$u_1 = a_1^3 a_2^{-5} a_3,$$

$$u_2 = a_4^{-3} a_5^{-2} a_6^{-2}.$$

Then

$$d_1 = min\{3, -2, -1\} = -2,$$

$$d_2 = min\{-3, -5, -7\} = -7$$

so $n_1 = 2^{-d_1} = 2^2$ and

$$(u_1^{-1}qu_1)^{2^2} = a_3^{-1} a_2^5 a_1^{-3} q^{2^2} a_1^3 a_2^{-5} a_3 = a_3^{-1} a_2^5 q^{2^5} a_2^{-5} a_3 = a_3^{-1} q a_3 = q^2 \in \langle q \rangle.$$

Similarly $n_2 = 2^{-d_2} = 2^7$ and

$$(u_2^{-1}qu_2)^{2^7} = a_6^2 a_5^2 a_3^3 q^{2^7} a_4^{-3} a_5^{-2} a_6^{-2} = a_6^2 a_5^2 q^{2^4} a_5^{-2} a_6^{-2} = a_6^2 q^{2^2} q_6^{-2} = q \in \langle q \rangle.$$

In the proof above we do not show that n_1 and n_2 are of minimal magnitude, even though they are. Because the natural order on the positive (negative) integers is a well ordering, we are guaranteed that integers of smallest magnitude exist so we can assume that m_1 and m_2 are said integers. Clearly

$(u_1^{-1}q_iu_1)^{m_1}, (u_2^{-1}q_iu_2)^{m_2} \in \langle q_i \rangle$ and $\langle q_i \rangle$ is cyclic so there exist smallest integers m_1' and m_2' such that $(u_1^{-1}q_iu_1)^{m_1'} = (u_2^{-1}q_iu_2)^{m_2'}$. This implies that every relation that holds in the group $\langle x \rangle * \langle y \rangle / \langle x^{m_1'}y^{-m_2'} \rangle^{\langle x \rangle * \langle y \rangle}$, also holds in $H = \langle u_1^{-1}qu_1, u_2^{-1}qu_2 \rangle$ via the homomorphism $x \to u_1^{-1}qu_1, y \to u_2^{-1}qu_2$. We now show that in fact these groups are isomorphic by showing that the relations of $\langle x \rangle * \langle y \rangle / \langle x^{m_1'}y^{-m_2'} \rangle^{\langle x \rangle * \langle y \rangle}$ are the only non-trivial relations that hold in H.

First note that the element $(u_1^{-1}qu_1)^{m_1'}$ is a power of both $u_1^{-1}qu_1$ and $u_2^{-1}qu_2$ so it generates a central subgroup of H that is identical to the one generated by $(u_2^{-1}qu_2)^{m_2'}$. For ease of notation let $x_1 = u_1^{-1}qu_1$ and $x_2 = u_2^{-1}qu_2$. If $R = e$ is a relation that holds in H, then we can express R as $R = v_1^{i_1}v_2^{i_2}...v_\beta^{i_\beta}$ where each i_j is a non-zero integer except i_1 and i_β either or both of which might be zero and such that $v_i = x_1$ if i is odd and $v_i = x_2$ if i is even. Note that if $\mid i_{2j+1} \mid \geq m_1'$ then we can rewrite R as $x_1^{\pm m_1'}v_1^{i_1}v_2^{i_2}...v_{2j+1}^{i_{2j+1} \pm m_1'}...v_\beta^{i_\beta}$ and similarly if $\mid i_{2j} \mid \geq m_2'$. Therefore we can assume that R has the form $q^\gamma v_1^{i_1}v_2^{i_2}...v_\beta^{i_\beta}$ where each $\mid i_{2j+1} \mid < m_1'$ and $\mid i_{2j} \mid < m_2'$.

We now apply Britton's Lemma to R. Either R is a power of q or we have a pinch or the form $a_j^{-\epsilon}qa_j^{\epsilon}$. We can continue to apply pinches until we have an expression equivalent to R written only in terms or powers of q, the powers of which sum to 0. But m_1' and m_2' are the smallest integral powers of $u_1^{-1}qu_1$ and $u_2^{-1}qu_2$ respectively, which lie in $\langle q \rangle$. Therefore, because each $\mid i_{2j+1} \mid < m_1'$ and $\mid i_{2j} \mid < m_2'$, we must have that they are all zeros; i.e., $R = q^\gamma$ and $\gamma = 0$. This proves that no other relations can hold in H.

Therefore,

$$H \cong \langle x \rangle * \langle y \rangle / \langle x^{m_1'} y^{-m_2'} \rangle^{\langle x \rangle * \langle y \rangle}$$

where $\langle x^{m_1'} y^{-m_2'} \rangle^{\langle x \rangle * \langle y \rangle} = \langle u^{-1} v u \mid u \in \langle x \rangle * \langle y \rangle, v \in \langle x^{m_1'} y^{-m_2'} \rangle \rangle$.

Thus H is an amalgamated free product which we show is right-orderable, by first considering the subgroup

$$I([H,H]) = \langle w \in H \mid \exists n \neq 0, w^n \in [H,H] \rangle$$

called the isolator of $[H,H]$. Naturally $H/[H,H]$ is abelian and $[H,H] \leq I([H,H])$ so $H/I([H,H])$ is an abelian group. Furthermore, $H/I([H,H])$ is torsion-free because if $wI([H,H])$ has finite order then $\exists i_1$ such that $w^{i_1} \in I([H,H])$ which implies that $\exists i_2$ such that $w^{i_1 i_2} \in [H,H]$ which means $w \in I([H,H])$. Therefore $H/I([H,H])$ is torsion-free abelian and hence right-orderable. To show that $I([H,H])$ is right-orderable, by virtue of Theorem 8, we need only show that it is a free group. Theorem 9 however, implies that $I([H,H])$ is free if

$$I([H,H])^H \bigcap \langle u_1^{-1} q u_1 \rangle = I([H,H])^H \bigcap \langle u_2^{-1} q u_2 \rangle = e$$

where $I([H,H])^H = \langle u^{-1} v u \mid u \in H, v \in I([H,H]) \rangle$. But $I([H,H])$ is a normal subgroup of H so $I([H,H])^H = I([H,H])$. Suppose there exists integer i_1 such that $(u_1^{-1} q u_1)^{i_1} \in I([H,H])$. Then by definition, there exists integer i_2 such that $(u_1^{-1} q u_1)^{i_1 i_2} \in [H,H]$. Therefore, $u_1 - 1 q u_1 \in I([H,H])$. But there exist integers m_1 and m_2 such that $(u_1^{-1} q u_1)^{m_1} = (u_2^{-1} q u_2)^{m_2}$ so $(u_2^{-1} q u_2)^{m_2} \in I([H,H])$ and thus $u_2^{-1} q u_2 \in I([H,H])$. This implies $H/I([H,H]) \cong e$ and

that H has no non-trivial abelian torsion-free quotients. But $H/[H,H] \cong \langle x, y \mid [x,y] = e, x^{m_1'} = y^{m_2'} \rangle$ by virtue of the isomorphism between H and $\langle x \rangle * \langle y \rangle / \langle x^{m_1'} y^{-m_2'} \rangle^{\langle x \rangle * \langle y \rangle}$. Thus $H/I([H,H])$ has an infinite cyclic subgroup and so an infinite cyclic quotient group which is a contradiction. Therefore, the supposition that there exists integer i_1 such that $(u_1^{-1} q u_1)^{i_1} \in I([H,H])$ is false and $I([H,H])$ is free and hence right-orderable.

This proves that every subgroup $\langle x_1, x_2 \rangle$ of $\langle q \rangle^A$ generated by two elements is right-orderable. We extend the proof to cover subgroups $\langle x_1, x_2, ..., x_i \rangle$, generated by i elements, by expressing H_j's iteratively as amalgamated free products of the first j generators. That is, given the subgroup

$$\langle x_1, ..., x_i \mid x_j = u_j^{-1} q u_j, u_j \in A \rangle$$

we express the subgroup generated by $\langle x_1, x_2 \rangle$ as

$$H_2 = \langle x_1 \rangle * \langle x_2 \rangle / \langle h_2 \rangle^{\langle x_1 \rangle * \langle x_2 \rangle} \ where \ h_2 = x_1^{m_1} x_2^{-m_2}$$

and in general we say

$$H_j = H_{j-1} * \langle x_j \rangle / \langle h_{j-1}^{m''_{j-1}} x_j^{-m_j} \rangle^{H_{j-1} * \langle x_j \rangle}.$$

Such a construction is always possible but may not yield a presentation of the intended group, unless the x_j's are first arranged in non-descending order with respect to the smallest positive integers k_i such that $x_i^{m_i} = q^{2^{k_i}}$ as the following example illustrates.

If $\langle x_1, x_2, x_3 \rangle = \langle u_1^{-1} q u_1, u_2^{-1} q u_2, u_3^{-1} q u_3 \rangle$ such that

$$u_1 = a_1^{-2} a_2^5,$$

$$u_2 = a_3^2 a_4^4,$$

$$u_3 = a_5^{-3} a_6^3.$$

Then finding n_1, n_2, and n_3 as before we have

$$(u_1^{-1} q u_1)^{2^2} = a_2^{-5} a_1^2 q^{2^2} a_1^{-2} a_2^5 = a_2^{-5} q a_2^5 = q^{2^5}$$

$$(u_2^{-1} q u_2)^1 = a_4^{-4} a_3^{-2} q a_3^2 a_4^4 = a_4^{-4} q^{2^2} a_4^4 = q^{2^6}$$

$$(u_3^{-1} q u_3)^{2^3} = a_6^{-3} a_5^3 q^{2^3} a_5^{-3} a_6^3 = a_6^{-3} q a_6^3 = q^{2^3}.$$

Now if we keep the order $x_1 = u_1^{-1} q u_1$, $x_2 = u_2^{-1} q u_2$, $x_3 = u_3^{-1} q u_3$ then

$$H_2 = \langle x_1 \rangle * \langle x_2 \rangle / \langle x_1^{2^3} x_2^{-1} \rangle^{\langle x_1 \rangle * \langle x_2 \rangle} \text{ and}$$

$$H_3 = H_2 * \langle x_3 \rangle / \langle (x_1^{2^3} x_2^{-1})^1 x_3^{-2^6} \rangle^{H_2 * \langle x_3 \rangle}.$$

But note that $x_3^{2^5} \neq x_1^{2^2}$ in H_3 but $(u_1^{-1} q u_1)^{2^2} = q^{2^5}$ and $(u_3^{-1} q u_3)^{2^5} = ((u_3^{-1} q u_3)^{2^3})^{2^2} = (q^{2^3})^{2^2} = q^{2^5}$ in $\langle u_1^{-1} q u_1, u_2^{-1} q u_2, u_3^{-1} q u_3 \rangle$.

However we can remedy this problem by taking $x_1 = u_3^{-1} q u_3$, $x_2 = u_1^{-1} q u_1$, and $x_3 = u_2^{-1} q u_2$. Then

$$H_2 = \langle x_1 \rangle * \langle x_2 \rangle / \langle x_1^{2^5} x_2^{-2^2} \rangle^{\langle x_1 \rangle * \langle x_2 \rangle} \text{ and}$$

$$H_3 = H_2 * \langle x_3 \rangle / \langle (x_1^{2^5} x_2^{-2^2})^2 x_3^{-1} \rangle^{H_2 * \langle x_3 \rangle} = \langle u_1^{-1} q u_1, u_2^{-1} q u_2, u_3^{-1} q u_3 \rangle$$

because H_3 has defining relations $x_1^{2^5} = x_2^{2^2}$ and $x_1^{2^6} = x_3$ which are precisely the defining relations of $\langle u_1^{-1} q u_1, u_2^{-1} q u_2, u_3^{-1} q u_3 \rangle$ under the above mapping.

It remains to show H_j is right-orderable. But this is done analogously to the two-generator subgroup case. $H_j / I([H_j, H_j])$ is torsion-free abelian and

hence right-orderable. By Theorem 1, if $I([H_j, H_j])$ is not a free group then there exists $w \neq e$ such that $w \in H_{j-1}$ or $w \in \langle x_j \rangle$ and $w \in I([H_j, H_j])$. To see that this is not possible, recall that every element of $\langle q \rangle^A$ has a power in $\langle q \rangle$ so every element of H_{j-1} and every element or $\langle x_j \rangle$ must also have a power in $\langle q \rangle$. Therefore if $w \in I([H_j, H_j])$ then some power of q is in $I([H_j, H_j])$ and thus every power or q in H_j is in $I([H_j, H_j])$. But then every element of H_j is in $I([H_j, H_j])$ since every element or H_j has a power which is a power of q. But $H_j \neq I([H_j, H_j])$ since H_j has an infinite cyclic quotient. Therefore $I([H_j, H_j])$ is a free group and hence right-orderable and hence so is H_j.

Therefore every finitely generated subgroup of $\langle q_i \rangle^A$ is right-orderable and therefore $\langle q_i \rangle^A$ itself is right-orderable for every $i \in \{1, 2, ..., \lambda\}$. But the groups $\langle t_i \rangle^A$ are completely analogous if we replace each u with u^{-1} in the above proof so each $\langle t_i \rangle^A$ is also right-orderable. Now $\langle l_i \rangle^A = \langle l_i \rangle$ which is infinite cyclic and so definitely right-orderable. Thus the free product of these groups is right-orderable so B is right-orderable. And A is free and so right-orderable so $G_2' = BA$ is right-orderable, and hence G_2^+ is also right-orderable. Finally $G_2 = G_2' * G_2^+$ so G_2 is right-orderable.

3.3.2 $A_{p_1 p_2}/K$ is right-orderable

Let

$$K = \langle [u, v] \mid u \in G_2, v \in P \rangle$$

where P is the free group $\langle p_1, p_2 \rangle$. Further, let H_1 be the subgroup generated by the diagonal elements, $x^{-1}x^+$ and x^+x^{-1} as x runs over the generators of G_2' and their inverses, and let H be the normal closure of H_1 in $A_{p_1p_2}$. By definition H is normal in $A_{p_1p_2}$. To see that K is also normal in $A_{p_1p_2}$ we note that $\forall g \in G_2, g^{-1}[u,v]g = [ug,v][g,v]^{-1} \in K$ and $\forall g \in P, g^{-1}[u,v]g = [u,g]^{-1}[u,vg] \in K$. Thus to right-order $A_{p_1p_2}$, we can simply right-order the groups $A_{p_1p_2}/HK$, HK/K, and K.

Note that $A_{p_1p_2}/HK$ is isomorphic to $P \times G_2'$ because $G_2' = G_2^+$ modulo H and elements of P and G_2 commute modulo K. As shown earlier, G_2 and G_2' are right-orderable, P is a free group of rank 2 and so also right-orderable, and so $A_{p_1p_2}/HK$ is right-orderable.

Now $HK/K \cong H/H \bigcap K$ is isomorphic to a subgroup of G_2 because the elements of H are conjugates of elements of G_2, which modulo K are only conjugated by elements of G_2, i.e. $\forall w \in \langle x^{-1}x^+ \rangle$, $\forall p \in \langle p_1, p_2 \rangle$, $p^{-1}wpK = wK$. Thus HK/K inherits from G_2 a right-order.

3.3.3 K is right-orderable

Finally, we show that K is just a free group of countable rank and thus also right-orderable.

Lemma 4 *The subgroup* $K = \langle [u,v] \mid u \in G_2, v \in P \rangle$ *of* $A_{p_1p_2}$, *is a free group of countable rank.*

The group $A_{p_1p_2}$ is finitely generated, and thus countable. $K \leq A_{p_1p_2}$ so it must be countably generated. To show K is free, we apply Britton's Lemma to the groups G_3, G_2, G_1, and G_0 in turn to show that no non-trivial relation of the form $W = e$ holds in K.

Beginning our proof by way of contradiction, assume we have

$$W = [u_1, v_1]^{n_1}[u_2, v_2]^{n_2}...[u_k, v_k]^{n_k} = e.$$

Applying Britton's Lemma in $K < G_3$ (recall $G_3 = A_{p_1p_2}$), it is not possible for the above presentation of W to be a $\Sigma_0 \bigcup \Sigma_1 \bigcup \Sigma_2$-word since each v_i is a Σ_3-word. Thus W contains a subword $p_j^{\epsilon} U p_j^{-\epsilon}$ where U is a word generated by

- $\{(A_i^+)^{-1} q_i^+ l_i^+,\ (t_i^+)^{-1}; 1 \leq i \leq \lambda\}$ if $p_j^{\epsilon} = p_1^{-1}$
- $\{A_i q_i^{-1} l_i^{-1},\ t_i; 1 \leq i \leq \lambda\}$ if $p_j^{\epsilon} = p_1$
- $\{B_i^{-1} t_i l_i,\ q_i^{-1}; 1 \leq i \leq \lambda\}$ if $p_j^{\epsilon} = p_2^{-1}$
- $\{B_i^+ (t_i^+)^{-1} (l_i^+)^{-1},\ q_i^+; 1 \leq i \leq \lambda\}$ if $p_j^{\epsilon} = p_2$

Let us consider how such a subword $p_j^{\epsilon} U p_j^{-\epsilon}$, also called a pinch, can occur in W.

One possibility is that a pinch could be completely contained in a single commutator $[u_i, v_i]$. In this case v_i must equal p_1, p_2, p_1^{-1} or p_2^{-1} because if one of the subwords $p_i^{\epsilon_1} U$ can be replaced, using the group relations, with a subword of the form $\overline{U} p_i^{\epsilon_1}$ then $p_j^{\epsilon_2} \overline{U}$ can not be replaced using the group

relations in a similar way, unless $p_i^{\epsilon_1} p_j^{\epsilon_2} = e$, which is trivial. A similar argument applies to subwords of the form $U p_i^{\epsilon_1}$. For example $p_1 t_i$ can be replaced by $(t_i^+)^{-1} p_i$ but $p_1 (t_i^+)^{-1}$ and $p_2^{\pm 1} (t_i^+)^{-1}$ can not be replaced using the group relations.

Therefore we have

$$[u_i, v_i] = u_i^{-1} v_i^{-1} u_i v_i = u_i^{-1} \gamma(u_i) v_i^{-1} v_i = u_i^{-1} \gamma(u_i)$$

where γ is the map defined by

$$\gamma : x^{-1} \mapsto x^+, \ x \in G_2'$$

$$\gamma : x^+ \mapsto x^{-1}, \ x \in G_2^+$$

The other possibility is that $v_i = v_{i+1}$ for some i and $n_i n_{i+1} < 0$, and the pinch occurs between $[u_i, v_i]$ and $[u_{i+1}, v_{i+1}]$. Assume, without loss of generality, that $n_i < 0$. We have

$$[u_i, v_i][u_{i+1}, v_{i+1}]^{-1} = u_i^{-1} v_i^{-1} u_i v_i v_{i+1}^{-1} u_{i+1}^{-1} v_{i+1} u_{i+1} = u_i^{-1} v_i^{-1} u_i u_{i+1}^{-1} v_{i+1} u_{i+1}$$

in which case $v_i \in \{p_1, p_2, p_1^{-1}, p_2^{-1}\}$ and

$$[u_i, v_i][u_{i+1}, v_{i+1}]^{-1} = u_i \gamma(u_i u_{i+1}^{-1}) u_{i+1}.$$

Thus, all of the pinches from G_3 yield subwords of the form

1. $u_i^{-1} \gamma(u_i)$
2. $u_i \gamma(u_i u_{i+1}^{-1}) u_{i+1}$

where each u_i in 1 and $u_i u_{i+1}^{-1}$ in 2, is generated by one of the bulleted sets above. Note that each pinch can use one and only one bulleted set in this way, but any and all of the bulleted sets may be used in different pinches throughout the expression W. We can continue applying Britton's Lemma until we produce a $\Sigma_0 \bigcup \Sigma_1 \bigcup \Sigma_2$-word equivalent to W. We label this equivalent expression W_2. Note that W_2 will be the product of subwords of the form of 1 and 2. Furthermore note that each such subword can be created in one and only one way, i.e. given the subword we can tell exactly what the pinch was and recover the original commutator or pair of commutators.

We again apply Britton's Lemma, this time to W_2 in G_2. In G_2, $\{l_i, l_i^+ \mid 1 \leq i \leq \lambda\}$ is the set of stable letters so either W_2 is a $\Sigma_0 \bigcup \Sigma_1$-word or W_2 has a pinch of the form

$$l_i^{-\epsilon} U l_i^{\epsilon}$$

or

$$(l_i^+)^{-\epsilon} U (l_i^+)^{\epsilon}$$

where U is a $\{a_j \mid 1 \leq j \leq n\}$-word for l_i and U is a $\{a_j^+ \mid 1 \leq j \leq n\}$-word for l_i^+. However there is no way to produce such a pinch using a product of words of the form of 1 and 2.

It is very easy to see that l_i^{ϵ} and $l_i^{+\epsilon}$ in the same u_i or $\gamma(u_i)$ or $\gamma(u_i u_{i+1}^{-1})$ can not form a pinch because they are necessarily separated by a word of the form $(A_i^+)^{-1} q_i^+$ or $A_i q_i^{-1}$ or $B_i^{-1} t_i$ or $B_i^+ (l_i^+)^{-1}$ and in order for $l_i^{-\epsilon} U l_i^{\epsilon}$ or $(l_i^+)^{-\epsilon} U (l_i^+)^{\epsilon}$ to be a pinch, we must have $U \in \langle a_i \rangle$ or $U \in \langle a_i^+ \rangle$ respectively.

Now consider the leftmost subword of the form 1. or 2. If the left most subword is of type 1, then any l_i^{ϵ} or $l_i^{+\epsilon}$ in u_i^{-1} is not part of a pinch and therefore not removable, since the pinches involving $\gamma(u_i)$ will not completely remove $\gamma(u_i)$ and any pinch involving letters of u_i^{-1} can not involve letters of $\gamma(u_i)$. The case for type 2 is the same for the l_i^{ϵ} or $l_i^{+\epsilon}$ in $\gamma(u_i u_{i+1}^{-1})$.

Therefore W_2 must already be a $\Sigma_0 \bigcup \Sigma_1$-word. But if the subwords of the form of 1 and 2, do not contain l_i^{ϵ} and $(l_i^{+})^{\epsilon}$, then they can not contain a_j's and a_j^{+}'s either, which are the stable letters of G_1.

Therefore, W_2 must be a Σ_0-word. But G_0 is a free group so W_2 must already be the identity. This means that the subwords of the form 1 and 2 in W_2 must cancel one another out, so one the subwords must be adjacent to its inverse. Since each of these subwords is created in a unique way, one of the commutators of W must be adjacent to its inverse, yielding a contradiction. We have shown that K is free and, therefore, right-orderable.

3.4 The Lattice-Ordered Group $L(A_{p_1p_2})$

We end this chapter by showing how to construct a lattice-ordered group $L(A_{p_1p_2})$ that may be a candidate to answer the question by A.M.W. Glass affirmatively. The author plans to investigate this possibility in the future. The method of embedding a right-ordered group into a lattice-ordered one is not new. We have shown that $A_{p_1p_2}$ is right-orderable, so taking the right regular representation of $A_{p_1p_2}$ yields a faithful homomorphism of $A_{p_1p_2}$ into

the group of order preserving permutations of the totally ordered set $A_{p_1p_2}$. To avoid confusion between the group $A_{p_1p_2}$ and the ordered set $A_{p_1p_2}$, we denote the latter Ω. Then for $g, h \in A_{p_1p_2}$, and $x \in \Omega$, we set $(g \vee h)(x) = max\{xg, xh\}$ and $(g \wedge h)(x) = min\{xg, xh\}$. This gives a lattice-ordered group generated by the generators of $A_{p_1p_2}$, and such that $A_{p_1p_2}$ is a subgroup. We denote this group $L(A_{p_1p_2})$.

It is important to note here that under the logical signature $\{e, \cdot, \wedge, \vee\}$ $L(A_{p_1p_2})$ is a finitely presented lattice-ordered group; the generators and defining relations of $L(A_{p_1p_2})$ are just those of $A_{p_1p_2}$. However, when viewed strictly as a group $L(A_{p_1p_2})$ is not even finitely generated. Thus, it could possibly be used to prove the existence of a group which is finitely presented in the class of lattice-ordered groups which have solvable word problem and unsolvable conjugacy problem. It would not be a group which is finitely presented as a group.

Chapter 4

The Embedding Question for Torsion-Free Groups

This chapter is devoted to giving an alternative proof to the following theorem, which partially answers a question by Collins.

Theorem 15 *Every torsion-free group G, with solvable power problem, can be embedded in a torsion-free group H, with solvable conjugacy problem.*

The result has already been established as a corollary to the following result

Theorem 16 (Olshanskii, Sapir [25]) *Every finitely generated group with solvable conjugacy problem is embeddable into a finitely presented group with solvable conjugacy problem. Moreover, every finitely generated recursively presented group G can be embedded into a finitely presented group H in such*

a way that the degree of unsolvability of the conjugacy problem in H coinsides with the degree of undecidability of the conjugacy problem in G.

However, to quote the authors,

"The construction in the proof of this theorem is complicated and employs ideas of three previous papers..." [25]

As our proof is short and self-contained we hope that it can be seen to have merit. The only previous result on which we rely here is the following well known theorem of H.N.N. and a simple construction that results from it.

Theorem 17 (theorem 2 of H.N.N. [14]) *Let μ_σ (where σ ranges over an index set Σ) be an isomorphism of a subgroup A_σ of a group G onto a second subgroup B_σ, not necessarily distinct from A_σ. Then there exists a group H containing G, and also containing a group T freely generated by a set of elements t_σ ($\sigma \in \Sigma$), such that for any σ in Σ the transform by t_σ of an element in A_σ is its image under μ_σ:*

$$t_\sigma^{-1} a_\sigma t_\sigma = \mu_\sigma(a_\sigma) \text{ for all } \sigma \in \Sigma \text{ and } a_\sigma \in A_\sigma$$

Collins' actual question, which remains open, can not be proven using the method we give here. We will explain this in the closing remarks of this chapter.

It was noted in [14] that H is torsion-free if G is so, because the only relations added are $t_\sigma^{-1} a_\sigma t_\sigma = \mu_\sigma(a_\sigma)$, which become trivial if we set the

elements of G equal to the identity; therefore the generators added in the extension do not satisfy a non-trivial relation on their own.

Following the construction outlined in [14], for each pair of distinct, non-identity elements (a, b) of G, we create an element $t_{(a,b,1)}$ such that $t_{(a,b,1)}^{-1} a t_{(a,b,1)}^{1} = b$. Because G is assumed to be torsion-free, all non-identity elements have the same (infinite) order, so this is possible. Then setting $G = G_0$ and $G_i = \langle G_{i-1}, t_{(a,b,1)} \mid a, b \in G_{i-1}, a \neq b \neq e \rangle$ we create the tower $G = G_0 \leq G_1 \leq ... \leq G_n \leq ...$ and set

$$H = \bigcup_{i=1}^{\infty} G_i.$$

It was shown in [14] that every two distinct non-identity elements of H are conjugate in H, and that if G_0 is countable and torsion-free, then so is H. This, however, does not imply that the conjugacy problem for H is solvable because, if H has an unsolvable word problem, then it must have an unsolvable conjugacy problem. However, if H has solvable word problem, and any two non-identity elements of H are conjugate, then we can solve the conjugacy problem in H. In order to determine if a and b are conjugate in H we need only determine whether or not one or both of a and b are the identity.

If G_0 has solvable power problem, and hence word problem, then, by induction, we will show that H has solvable power problem. Assume $n \geq 1$ and G_{n-1} has solvable power problem. We begin creating a normal form for elements of G_n by deleting as many occurrences of generators of $G_n \backslash G_{n-1}$

as possible. Although this normal form will not be unique, in that a given element may be expressed by more than one word in this form, we will have the result that an element reduced to such a normal form is the identity only if it contains no occurrences of generators (or inverses of generators) of $G_n \backslash G_{n-1}$. This implies that the word problem will then be reduced to the word problem in G_{n-1}, which, by inductive assumption, is solvable. From now on we refer to the occurrence of a subword of the form $t_{a,b,n}^{\pm 1}$, in a word w as a singleton subword of $G_n \backslash G_{n-1}$.

Let w be a word in the generators and their inverses of G_n. If w contains no singleton subwords of $G_n \backslash G_{n-1}$ we are done. Otherwise, picking each pair of consecutive singleton subwords of w that lies in $G_n \backslash G_{n-1}$ we get a subword of the form

$$(*)\ t_{a,b,n}^{\epsilon_1} w' t_{c,d,n}^{\epsilon_2}, \qquad where\ \epsilon_i = \pm 1$$

We begin by freely reducing w.

Recall that $t_{a,b,i}^{-1} a t_{a,b,i}^{1} = b$ so $t_{a,b,i}^{1} b t_{a,b,i}^{-1} = a$. Thus, with each $(*)$; we do the following.

1. Determine if $t_{a,b,n} = t_{c,d,n}$. If not, we move to the next $(*)$, otherwise we proceed to step 2.

2. If $\epsilon_1 = 1$ and $\epsilon_2 = -1$, determine if $w = b^k$ for some integer k less than or equal to the length of w'. Both w' and b^k lie in G_{n-1} so we can do this in a finite number of steps. If $w' = b^k$, replace $(*)$ with a^k. If $\epsilon_1 \neq 1$ or $\epsilon_2 \neq -1$ or $w' \neq b^k$ do nothing. Move to step 3.

3. If $\epsilon_1 = -1$, $\epsilon_2 = 1$, determine if $w' = a^k$. If so replace $(*)$ with b^k; otherwise do nothing. Move to the next (*).

We then make successive passes by repeating the above algorithm, freely reducing after each pass, until we complete a pass where no singleton subwords have been deleted. Note that w has only a finite number of such singleton subwords of $G_n \backslash G_{n-1}$, so we need to make only finitely many passes.

There is no other way to delete a singleton subword of $G_n \backslash G_{n-1}$ so if any remain, w can not be the identity element. Otherwise w is reduced to an expression solely in the generators and inverse of generators of G_{n-1}. By our inductive assumption, G_{n-1} has solvable word problem, so we can tell whether or not this normal form of w is equal to the identity. This proves the following lemma.

Lemma 5 *In the above construction, if G_0 is a torsion-free group and G_{n-1} has solvable power problem, then G_n has solvable word problem.*

Note that the above lemma does not imply, by induction or otherwise, that H has solvable word problem and therefore solvable conjugacy problem. However if we can show that G_n also has solvable power problem, then the result will follow by induction.

Given w_1, w_2 in G_n, it remains to show whether or not we can determine if there exists an integer k such that $w_1 = w_2^k$. Since G_n was shown to have solvable word problem, we may assume that $w_1 \neq w_2$.

We begin by applying the above algorithm to w_1 and w_2 to delete any deletable singleton subwords of G_n. This gives

$$w_1 = u_1 v_1 u_2 v_2 ... u_m v_m$$

$$w_2 = r_1 s_1 r_2 s_2 ... r_L s_L$$

where u_i, $r_i \in \langle G_n \backslash G_{n-1} \rangle$ and s_i, $v_i \in G_{n-1}$.

Define $\overline{w_i} = \phi(w_i)$ where $\phi : G_n \rightarrow G_n / G_{n-1}^{G_n}$, the factor group being isomorphic to a free group. The homomorphism essentially sets each $s_i = v_i = e$.

The power problem in $G_n / G_{n-1}^{G_n}$ is easily solvable because $G_n / G_{n-1}^{G_n}$ is a free group. If for some k, we have $\overline{w_1} = \overline{w_2}^k$ then we need only check whether or not $w_1 = w_2^k$ for that particular k. This is so because if there exists some k such that $w_1 = w_2^k$, then certainly $\overline{w_1} = \overline{w_2}^k$ must also be true. Since G_n has solvable word problem, for fixed k we can check whether or not $w_1 = w_2^k$ is true.

The only case that remains occurs when $\overline{w_1} = \overline{w_2} = e$. In this case the $t_{(a,b,n)}^{\epsilon}$ are 'balanced' in the sense that for every occurrence of the singleton subword $t_{(a,b,n)}^{\epsilon}$ in w_2 (w_1) there is a corresponding occurrence of $t_{(a,b,n)}^{-\epsilon}$. We therefore assume that w_1 and w_2 have been reduced to a normal form by applying the above algorithm and proceed by induction on the number m of such pairs of singleton subwords of $G_n \backslash G_{n-1}$ in w_2.

If w_2 has no pairs of singleton subwords of $G_n \backslash G_{n-1}$, then either w_1 has pairs of singleton subwords of $G_n \backslash G_{n-1}$ or it does not. If it does, then there

is no integer k such that $w_1 = w_2^k$, or w_1 has no such pairs. If w_1 has no such pairs, then both w_1 and w_2 lie in G_{n-1} and we can solve the power problem for them by the inductive assumption that G_{n-1} has solvable power problem.

If w_2 has one pair of singleton subwords of $G_n \backslash G_{n-1}$, then

$$w_2 = v_1 t_{(a,b,n)}^{\epsilon} v_2 t_{(a,b,n)}^{-\epsilon} v_3$$

where v_1, v_2, $v_3 \in G_{n-1}$. Without loss of generality we treat only the case where $\epsilon = -1$. Three cases arise by applying the above reduction algorithm to w_2^2 to see if the number of non-deletable pairs of singleton subwords of $G_n \backslash G_{n-1}$ increases, decreases or remains constant at one.

For case one, if the number of singleton subwords increases, then w_2^2 has two pairs of singleton subwords and has the normal form

$$w_2^2 = v_1 t_{(a,b,n)}^{-1} v_2 t_{(a,b,n)} v_4 t_{(a,b,n)}^{-1} v_2 t_{(a,b,n)} v_3$$

where v_4 is the freely reduced form of $v_3 v_1$. Then clearly for every integer k, w_2^k has k pairs of singleton subwords of $G_n \backslash G_{n-1}$. Therefore we need only count the number of pairs, say k, of singleton subwords of w_1 and test $w_1 = w_2^k$ for this value of k only. We can do so because the previous lemma shows that the word problem in G_n is solvable.

For the second case, if w_2^2 has fewer singleton subwords of $G_n \backslash G_{n-1}$ than w_2, then obviously $w_2^2 \in G_{n-1}$. In this case, if $w_1 = w_2^{2k}$ for some integer k then $w_1 \in G_{n-1}$ and if $w_1 = w_2^{2k+1}$ then $w_1 w_2^{-1} \in G_{n-1}$. Therefore, to solve this case we simply check whether or not w_1 or the reduced form of $w_1 w_2^{-1}$ are in G_{n-1}. If either is, we solve the corresponding power problems in G_{n-1}.

For the final case, if w_2^2 has exactly one pair of singleton subwords of $G_n \backslash G_{n-1}$ then

$$w_2^2 = v_1 t_{(a,b,n)}^{-1} v_2^2 t_{(a,b,n)} v_3$$

if $v_1 = v_3^{-1}$ or

$$w_2^2 = v_1 t_{(a,b,n)}^{-1} v_2 a^j v_2 t_{(a,b,n)} v_3$$

if for some integer j, $v_3 v_1 = b^j$. In either case, for any integer k we have respectively

$$w_2^k = v_1 t_{(a,b,n)}^{-1} v_2^k t_{(a,b,n)} v_3$$

or

$$w_2^k = v_1 t_{(a,b,n)}^{-1} v_2 (a^j v_2)^k t_{(a,b,n)} v_3.$$

In both cases we use the above reduction algorithm to delete as many singleton subwords of $G_n \backslash G_{n-1}$ of $t_{(a,b,n)} v_1^{-1} w_1 v_3^{-1} t_{(a,b,n)}$ as possible and check whether or not it is an element of G_{n-1}. The power problem now reduces to that of G_{n-1}.

We now assume that if w_2 has fewer than m pair of singleton subwords of $G_n \backslash G_{n-1}$ then the power problem for w_1 and w_2 is solvable. For w_2 having exactly m such pairs, we again consider the same three cases of the reduced form of w_2^2.

The first case is completely analogous to case one above so we omit the details. In the second case, if w_2^2 has fewer singleton subwords of $G_n \backslash G_{n-1}$ than w_2, then by our inductive hypothesis, we can solve the following questions that together, constitute the power problem in this case.

1. Does there exist an integer k such that $w_1 = w_2^{2k}$?

2. Does there exist an integer k such that $w_1 w_2^{-1} = w_2^{2k}$?

The final case is also completely analogous to the third case above and so is left to the reader. This completes the proof theorem 1.

Recall that D.J. Collins asked if every torsion-free group with solvable word problem could be embedded in a group with solvable conjugacy problem. This stronger result can not be derived from the above method because the word problem in G_n is equivalent to the power problem of G_{n-1} and therefore implies a solution to the power problem in G_0.

Chapter 5

An application of the Conjugacy Problem

5.1 The Braid Cryptosystem

Commercial applications are by no means necessary to justify the study of any area of mathematics, this being particularly true of areas of pure mathematics, but they are always greeted warmly when they appear. In this chapter we present a public key cryptosystem which, utilizing groups with unsolvable conjugacy problem, significantly improves the security of the Braid Cryptosystem, while maintaining what appears to be comparable efficiency.

A cryptosystem, i.e. a method for securely exchanging secret information, is called a public key, or asymmetric, cryptosystem if it is believed that the decoding key can not be deduced from the encoding key in a reasonable

amount of time with available and hopefully even with future computing technology. This allows for the encryption key to be made public without compromising the cryptosystem.

The Braid Cryptosystem is a public key cryptosystem developed primarily by Ko, Lee, Cheon, Han, Kang and Park [15]. The braid group on $n+1$-strands has the following presentation:

$$B_n = \langle \sigma_1, ..., \sigma_n \mid \ \sigma_i\sigma_j = \sigma_j\sigma_i \ if \ |i-j| > 1, \ \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j \ if \ |i-j| = 1 \rangle.$$

where σ_i can be represented graphically as a crossing of strand i over strand $i+1$ and for use in the cryptosystem we assume that n is even.

Before describing the Braid cryptosystem we first point out the properties of the braid group that make the cryptosystem possible. Note that the first set of defining relations imply that any element from $B_L = \langle \sigma_1, ..., \sigma_{n/2-1} \rangle$ commutes with any element of $B_R = \langle \sigma_{n/2+1}, ..., \sigma_n \rangle$. Another important feature of the Braid group stems from the fact that it is an automatic group. We will discuss automatic groups in more detail in the next section, but for now it is sufficient to note that the word problem for any automatic group can be solved in quadratic time. This means that for any automatic group G, there exists a positive constant c such that given any word w of length n in the generators and their inverses, we can determine algorithmically in cn^2 steps or fewer whether or not $w = e$ in G. Finally it is important that the word problem in automatic groups is solved in quadratic time by converting any element of the group to its unique normal form in quadratic time and

then comparing if the normal form is identical to e.

Cryptosystems are conventionally explained in terms of two parties Alice and Bob exchanging secret information via the cryptosystem. Following this convention, the Braid cryptosystem works as follows.

1. Alice chooses an element $x \in B_n$ and $a \in B_L$, computes the normal form of $c = a^{-1}xa$ and passes c and x to Bob, retaining a as her secret.

2. Bob chooses an element $b \in B_R$, computes the normal form of $d = b^{-1}xb$ and passes it to Alice retaining b as his secret.

3. Alice computes $a^{-1}da$, maps it via a hashing function h to a binary word and then encodes her binary message m as $m' = h(a^{-1}da) \oplus m$ to send to Bob. Here $\cdot$ denotes the 'exclusive or' function on binary strings.

4. Bob then computes $h(b^{-1}cb) \oplus m'$. Since a and b commute, $a^{-1}da = b^{-1}cb$ and $h(b^{-1}cb) \cdot m' = m$

According to its creators in [15], the beauty of this algorithm lies in it's efficiency to encode and decode the messages, and the limited memory and CPU capacity needed to run it: making it ideal for use in cell phones, pda's, etc.

The success of the braid cryptosystem depends on Alice and Bob keeping a and b secret respectively. This is achieved if given $a^{-1}xa$ and x, it is algorithmically infeasible to a and analogously for b. By 'algorithmically in-

feasible' we usually mean that the solution should take at least exponential time in the sum of the lengths of words x and $a^{-1}xa$. The fastest known algorithm for solving the conjugacy problem in biautomatic groups is exponential; however there have been numerous attacks on the conjugacy problem of the braid cryptosystem which take only polynomial time in most cases. This requires Alice and Bob to be very selective in their choice of a, b, and x, which requires additional time and lowers the safety against a brute force attack. In fact, the method for selecting them to ensure safety against the attack detailed in [20], is directly at odds with the method for selecting them to ensure safety against brute force attacks. In 1992, the authors of [9], on page 204, state "there is probably a polynomial time algorithm [which solves the conjugacy problem in the braid groups] using pseudo-Anosov homotopies." The recent paper [10] proves the existence of a polynomial time algorithm which solves the Diffie-Helman conjugacy problem in the braid group. The paper concludes that the cryptosystem is no longer secure.

However, if we could apply the same or similar cryptographic algorithm to a group that had all of the positive aspects of the braid group detailed above but also had an unsolvable conjugacy problem, then the resulting cryptosystem would be as efficient as the braid cryptosystem at encoding and decoding messages and, at the same time, would be much more secure because it is difficult to even conceive of a an attack other than brute force against a problem that can be proven to be algorithmically unsolvable. It should be noted that even if the problem is unsolvable, by limiting the length

of a possible solution we can always test all possibilities of equal or lesser length to see if a solution exists. This is, by definition, a brute-force attack. In fact as explained at the end of this chapter, if a problem is algorithmically unsolvable but solutions for the problem can be confirmed in polynomial time, the problem has new potential for proving that $P \neq NP$.

In the sections 5.2, 5.3, and 5.4, we detail properties of automatic, biautomatic and hyperbolic groups. In the section 5.5 we use these results to give an explicit example, with proof, having all of the necessary properties of the braid group but also having unsolvable conjugacy problem. In section 5.6, we give two new algorithms detailing cryptosystems based on this new group. In the final section, 5.6, we discuss the security of these new cryptosystems.

5.2 Automata theory

As this is probably the area that the reader is least familiar with we will try to give a less formal introduction. Most of the definitions and theorems are stolen shamelessly from [9] and [2], but discussions are not.

First we make the following definitions.

Definition 1 *[9] The set of all strings over an alphabet A is denoted A^*.*

Definition 2 (language [9]) *A language over A is a subset of A^*, together with the alphabet A. Mention of the alphabet A is frequently suppressed. Nevertheless, if we are being rigorous, we must distinguish between the null language over the alphabet$\{x\}$ and the null language over the alphabet $\{x, y\}$.*

When theoretical computer science was in its infancy, the question arose as to which languages over a finite alphabet could be recognized or accepted by a computer. The definition of 'accepted' will be formalized shortly, but for now we simply mean that a computer accepts a language L over an alphabet A if given any word w in A^* it can determine in a finite number of steps whether or not $w \in L$. The notion of such a computer will also be formalized shortly but for now we note only that its behavior must be completely explainable by a finite number of rules, which take up a finite number of words and symbols; therefore the number of such computers must be countable. Now, if A has at least one element, then A^* is countably infinite and the number of languages on A must be uncountable. So not every language can be accepted by a computer.

Several constructs such as Turing machines, context sensitive grammars, context free grammars, and automata were created (in the sense that a precise definition of what rules they could follow and what behaviors they could perform) which accept different types of languages. It turns out that automata are the most restrictive in that the subset of languages that they accept is the smallest. However there are computational advantages to automata because the languages that they accept are in a sense much better behaved. For comparison between these types of languages and computers we refer the reader to any introductory text in the theory of formal languages. For the remainder of the section we restrict ourselves to automata and the languages they accept.

Languages recognized by automata are said to be regular. As the definition of a regular language is often easier to comprehend than that of a finite state automaton, we give the definition of a regular language first. It is hoped that the tedious definition of finite state automata will be made more bearable with the motivation that it is the means for a computer to accept the very logical construction, i.e. the regular language. To this end we first give the definition of a regular expression.

Definition 3 (regular expression [9]) *A regular expression over an alphabet A is a particular type of string (specified below) over the alphabet E formed by adjoining to A the following five symbols, which are assumed not to lie in A already: (,), *, +, and* ϵ

We pronounce + as "or", and * as "star". Informally, parentheses are used for grouping, * denotes repetition, + is used to combine alternative patterns, and ϵ is the null string. A regular expression over A can be seen as a string which defines a subset of $A*$. By the regular expression of a word of $A*$, we mean a regular expression which it satisfies.

Above, we gave the definition of A^* where A is a finite alphabet but the immediately preceding definition requires the use of the * operator on words and languages. The next definition explains this concept.

Definition 4 (concatenation of languages [9]) *If K and L are languages over the same alphabet A, we define their concatenation KL to be the set of*

strings w for which $w = w_1w_2$ in A^*, where $w_1 \in K$ and $w_2 \in L$. If K or L is empty, so is KL. We define the star closure of K as

$$K^* = \bigcup_{n \geq 0} K^n,$$

where $K^0 = \{\epsilon\}$ and $K^n = K^{n-1}K$ for $n > 0$.

A regular language can now be defined as any language that is the set of all words over an alphabet A that have the same regular expression. In order for this definition to be rigorous we need either that regular expressions are unique or a method for determining the equality of two regular expressions. The first condition cannot be satisfied since we can always add extra parentheses without effect. However, the following rules allow us to compare any two regular expressions and determine if they are equal. We denote by $L(r)$ the language defined by the regular expression r.

- $L((r)) = L(r)$
- $L(r^*) = (L(r))^*$
- $L(r_1r_2) = L(r_1)L(r_2)$

With these languages in mind we now give a characterization of the machines which recognize them.

Definition 5 (finite state automaton [9]) *A finite state automaton(or simply automaton) is a quintuple (S, A, μ, Y, s_0), where S is a finite set, called*

the state set, A is a finite set, called the alphabet, $\mu : S \times A \to S$ is a function, called the transition function, Y is a (possibly empty) subset of S called the subset of accept states, and $s_0 \in S$ is called the start state or the initial state.

There are generalizations of the finite state automaton which turn out to be very useful in understanding examples of the original definition because the generalizations are easier to manipulate and by the following theorem, are equivalent.

Theorem 18 (Kleene, Rabin, Scott [9]) *Let A be a finite alphabet. The following four conditions on a language over A are equivalent:*

1. *The language is recognized by a deterministic finite state automaton.*
2. *The language is recognized by a non-deterministic finite state automaton.*
3. *The language is recognized by a generalized finite state automaton.*
4. *The language is defined by a regular expression.*

Definition 6 (non-deterministic finite state automaton [9]) *A non-deterministic finite state automaton is a quintuple (S, A, μ, Y, S_0), where A is a finite set, called the alphabet, S_0 is a subset of S called the subset of initial states, Y is a subset of S called the subset of accept states, and μ is a set of arrows with labels in the enlarged alphabet $A \cup \{\epsilon\}$. The symbol ϵ is assumed not to lie in A and it is meant to denote the null string.*

A deterministic finite state automaton can be considered a special case of a non-deterministic finite state automaton for which the following conditions are satisfied.

- There are no arrows labeled ϵ.
- Each state is the source of exactly one arrow with any given label from A.
- The subset S_0 has exactly one element.

The convention of a non-deterministic finite state automaton is useful in proving that a language is regular because non-deterministic finite state automata are often easier to construct than their deterministic equals.

Definition 7 (generalized finite state automaton [9]) *A generalized finite state automaton is the same as a non-deterministic finite state automaton, except that each arrow is labeled by a regular expression over A.*

Generalized finite state automata are useful to determine the language that a standard automaton recognizes because we can use intermediate generalized automata to convert the deterministic or non-deterministic automaton to a regular expression.

We end this section with the theorem from [2] which will be useful to the original section of this chapter.

Theorem 19 *[2] Suppose K and L are regular sets contained over A. Then the following hold:*

1. *A finite subset of A^* is a regular language.*

2. *$A^* \backslash K$ is a regular language.*

3. *$K \cup L$ is regular.*

4. *$K \cap L$ is regular.*

5. *KL is regular.*

6. *K^* is regular.*

7. *A^* and the empty set are regular.*

8. *If B is a second finite set and ϕ is a homomorphism of the monoid A^* into the monoid B^*, then $\phi(L)$ is regular over B.*

9. *If ϕ is a homomorphism of A^* into B^* and if J is a regular subset of $B*$, then $\phi^{-1}(J)$ is regular over A.*

5.3 many-variable regular languages

In order to define the concept of an automatic group we need to deal with ordered pairs of elements of a regular language, so we first consider the generalization to sets of n-tuples of elements.

Definition 8 (many-variable language [9]) *Let $A_1, ..., A_n$ be alphabets. By a language over $(A_1, ..., A_n)$ we mean a set of n-tuples of strings $(w_1, ..., w_n)$*

where $w_i \in A_i^*$, together with the n-tuples of alphabets $(A_1, ..., A_n)$. A language over an n-tuple of alphabets is called an n-variable language.

This definition allows us to consider languages which are subsets of $A_1^* \times ... \times A_n^*$. However these languages may not be generated by subsets of $A_1 \times ... \times A_n$ because there are no null-elements in the alphabets A_i. The following two definitions alleviate this problem.

Definition 9 (padding [9]) *Let $A_1, ..., A_n$ be alphabets. We adjoin to each A_i an end-of-string or padding symbol, denoted by $\$_i$, which is assumed not to lie in A_i, and we define $B_i = A_i \cup \{\$_i\}$. The padded alphabet associated with $(A_1, ..., A_n)$ is the set*

$$B = B_1 \times ... \times B_n \backslash (\$_1, ..., \$_n).$$

Definition 10 (padded extension [9]) *Given a language L over $(A_1, ..., A_n)$, we define a one-variable language $L^\$$ over the padded alphabet B associated with $(A_1, ..., A_n)$, as follows: For each n-tuple $(w_1, ..., w_n) \in L$, let m be the maximal length of the w_i, for $1 \leq i \leq n$. We pad each w_i with $\$_i$'s at the end so as to make its length m. The resulting n-tuple of strings is a string of length m in the alphabet B; these are the strings of $L^\$$. We call $L^\$$ the padded extension of L.*

We can now define a regular many-variable language.

Definition 11 (regular many-variable language [9]) *We say that L is a regular language over $(A_1, ..., A_n)$ if $L^\$$ is a regular language over the padded*

alphabet B associated with $(A_1, ..., A_n)$. A finite state automaton over B accepting the language $L^\$$ is said to be an n-variable automaton over $(A_1, ..., A_n)$ accepting L.

We end this section with, in some sense, the analog of the final theorem of the previous section.

Theorem 20 ([9]) *Let L and L' be regular languages over $(A_1, ..., A_n)$.*

1. *The languages $\neg L$, $L \cup L'$ and $L \cap L'$ are regular languages over $(A_1, ..., A_n)$.*
2. *For any alphabet A_{n+1}, the language*

$$\{(w_1, ..., w_n, w_{n+1}) \mid (w_1, ..., w_n) \in L\}$$

is a regular language over $(A_1, ..., A_{n+1})$.

3. *For any permutation σ of $\{1, ..., n\}$, the language*

$$L_\sigma = \{(w_1, ..., w_n) \mid (w_{\sigma(1)}, ..., w_{\sigma(n)}) \in L\}$$

is a regular language over $(A_{\sigma(1)}, ..., A_{\sigma(n)})$.

5.4 automatic groups

With the previous definition we can now define the concept of an automatic group.

If $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ is a finite set and G a finitely generated group such that the map $x \mapsto \overline{x} (x \in \mathcal{X}, \overline{x} \in G)$ from $\mathcal{X}$ into the generators of G, and

their inverses, extends to a surjective homomorphism π from some regular subset (language) L of $\mathcal{X}^*$ to G, then G is said to be an automatic group if the following subsets of $v(\mathcal{X}^* \times \mathcal{X}^*)$ are also regular:

- $L_= = \{v(w_1, w_2) \mid w_1, w_2 \in L, \overline{w_1} = \overline{w_2}\}$
- $L_{x_i} = \{v(w_1, w_2) \mid w_1, w_2 \in L, \overline{w_1} = \overline{w_2 x_i}\}$

In the above definition v is the map defined as follows. If $w_1 = a_1...a_n$, $w_2 = b_1...b_m$ then

- $v : (w_1, w_2) \mapsto (a_1, b_1)(a_2, b_2)...(a_m, b_m)(a_{m+1}, \$)...(a_n, \$)$ if $m < n$.
- $v : (w_1, w_2) \mapsto (a_1, b_1)(a_2, b_2)...(a_n, b_n)$ if $m = n$.
- $v : (w_1, w_2) \mapsto (a_1, b_1)(a_2, b_2)...(a_n, b_n)(\$, b_{n+1})...(\$, b_m)$ if $m > n$.
- $v : (e, e) \mapsto e$.

Here \$ is used as a padding symbol to allow for the comparison of two strings of $\mathcal{X}^*$ of different length.

Another way to characterize automatic groups is via a geometric interpretation using their so-called Cayley graphs.

Definition 12 (Cayley graph [9]) *For a group G with generating set A, the Cayley graph of G relative to A is a directed graph where the vertices are the elements of G and the edges are the set of triples $(g, a, \overline{ga})$, where $a \in A$ and $g \in G$. For each such edge, g is called the base point and $\overline{ga}$ the terminus.*

An alternate definition of automatic groups is that with respect to some inverse closed generating set A, two travelers proceeding at the same speed along the words w_1 and w_2 from the same base point in the Cayley graph will always remain a bounded distance apart. This is the k-fellow traveler property. More formally, for a word $w \in A^*$, we denote the length of w by l(w), and for $g \in G$, $l(g)$ denotes the length of the shortest word $w \in A^*$ such that $\overline{w} = g$. The term $w(t)$ denotes the prefix of w of length t when $t \leq l(w)$ and $w(t) = w$ if $t > l(w)$. Then we have the following definition.

Definition 13 (K-fellow traveler property [9]) *If for language L on alphabet A, the group G has the map $x \rightarrow \overline{x}$ defined above and there exists some constant k such that for every $w_1, w_2 \in L$ such that $\overline{w_1 a} = \overline{w_2}$ for some $a \in A$ we have $l(\overline{w_1^{-1}(t) w_2(t)}) \leq k$ then G has the k-fellow traveler property (with respect to A and L.)*

It turns out that

Theorem 21 ([9]) *A group G has automatic structure (A, L) if and only if L is a regular language and G has satisfies the k-fellow traveler property for some k.*

There are several ways to improve on the automatic structure of a group. That is we can impose stricter conditions to gain further properties. Among them are shortlex automatic groups, biautomatic groups and word hyperbolic groups. Let A be an ordered alphabet. Recall that lexicographic order ranks

the strings of the same length in A^*, by comparing the letters in the first position where the strings differ.

Definition 14 (Shortlex order [9]) *For a set or language L on an alphabet A, the shortlex order is defined by $w < v$ if $l(w) < l(v)$ or $l(w) = l(v)$ and w comes before v in the lexicographical order. Note that this is a well ordering.*

A string $w \in A^*$ is called a geodesic if it has minimal length among all strings representing the same element of G or equivalently if it is the shortest path between to fixed points of the Cayley graph of G. The language of all geodesic strings maps finite-to-one onto G, but in general this language does not have to be regular or even recursively enumerable.

Definition 15 (strongly geodesically automatic [9]) *If the language of all geodesics L' is part of an automatic structure(A, L') for G, we say that G is strongly geodesically automatic.*

Definition 16 (weakly geodesically automatic [9]) *If some language consisting of only geodesics is part of an automatic structure for G, we say that G is weakly geodesically automatic.*

A string $w \in A^*$ is a shortlex-geodesic if it is minimum in the shortlex order among all strings representing the same element of G as w.

Definition 17 ([9]) *If the language of shortlex-geodesics is part of an automatic structure, we say the group is shortlex automatic.*

Theorem 22 ([9]) *Let G be an automatic group with automatic structure (A, L). Let $L' \subset L$ be the set of all strings $w \in L$ such that w is shortlex-minimal in $L \cap \pi^{-1}(\overline{w})$. Then (A, L') is an automatic structure for G. In particular, G has an automatic structure over A with the uniqueness property.*

The uniqueness property mentioned above is simply that (A, L) has the uniqueness property if $\pi : L \to G$ is one-to-one. An automatic structure (A, L) is prefix closed if every prefix of a word in L is also a word in L.

Theorem 23 ([9]) *A shortlex automatic structure (A, L) for a group G is necessarily prefix closed and has the uniqueness property.*

An even stronger type of automaticity is biautomaticity.

Definition 18 (biautomatic [9]) *Let G be an automatic group with automatic structure (A, L) where A is closed under the inversion. We say that the structure is biautomatic if (A, L^{-1}) is also an automatic structure.*

Theorem 24 (biautomatic implies solvable conjugacy problem [9]) *If G has a biautomatic structure, the conjugacy problem is solvable in G, that is, one can algorithmically determine whether or not two words represent conjugate elements in G.*

It should be noted that so far the fastest such algorithm takes a time which is exponential in the length of the elements.

However there is a family of groups for which the word problem can be solved in linear time and the conjugacy problem in $O(nlogn)$. We refer of

course to word hyperbolic groups. A formal definition of word hyperbolic groups can be found in [2] and [9] but a basic definition is that there exists a constant δ such that any triangle in the Cayley graph of the group, consisting of geodesic sides must have an area less than δ. It turns out that calculating this δ is necessary for the algorithm for the word and conjugacy problems but it is difficult to calculate in general. We can however, calculate it when the group is the fundamental group of a two dimensional surface, which will be important in subsequent sections of this chapter.

It turns out that word hyperbolic groups are strongly (and hence weakly) geodesically automatic with respect to any ordering of the generating set. They are also biautomatic with respect to any generating set.

We end this section by stating the definition of 'Turing machine' that we will adopt as convention. In this thesis, by a 'Turing machine' we always refer to a finite list of four-tuples which define the actions of a theoretical computer. The the theoretical computer has a finite number of states $q_0, , q_n$, and a finite number of tape symbols $s_0, , s_m$. It is assumed that q_0 is the halting state which, when entered, forces the computer to stop. The types of four tuples are $(q_{j_1}, s_{j_2}, s_{j_3}, q_{j_4})$, $(q_{j_1}, s_{j_2}, R, q_{j_4})$, and $(q_{j_1}, s_{j_2}, L, q_{j_4})$. The first tells the computer, when in state q_{j_1} and reading symbol s_{j_2}, to write symbol s_{j_3} and enter state q_{j_4}. The second four tuple tells the computer, when in state q_{j_1} and reading symbol s_{j_2}, to move right one space and enter state q_{j_4}. The third four-tuple, is the same as the second except that it tells the computer to move one space to the left instead of to the right.

5.5 A candidate for a new cryptosystem

In this section we modify a group, proven in [1] to be biautomatic and have a subgroup with unsolvable conjugacy problem, to produce a group that retains these properties but for which the subgroup with unsolvable conjugacy problem also has commuting subgroups. Further we will show that the normal form provided by the automatic structure provides a security level that greatly exceeds that of the braid cryptosystem.

As in [11], we begin with a Turing machine T with unsolvable halting problem, tape alphabet $s_o, s_1, ..., s_{M-1}$ and internal states $q_0, q_1, ..., q_N$ with q_1 as the start state and q_0 as the unique halting state. Markov and Post, are credited in [11] with the following construction of a finitely presented semigroup $\gamma(T)$ with unsolvable word problem

$$\gamma(T) = \langle h, s_0, s_i, ..., s_{M-1}, q, q_1, ..., q_N \mid R(T) \rangle$$

where the relations $R(T)$ are

$$q_i s_j = q_l s_k \quad if \quad q_i s_j s_k q_l \in T$$

and for all $b \in \{0, 1, ..., M-1\}$:

$$q_i s_j s_b = s_j q_l s_b \quad if \quad q_i s_j R q_l \in T,$$

$$q_i s_j s_M = s_j q_l s_0 s_M \quad if \quad q_i s_j R q_l \in T,$$

$$s_b q_i s_j = q_l s_b s_j \quad if \quad q_i s_j L q_l \in T,$$

$$s_M q_i s_j = s_M q_l s_0 s_j \quad if \quad q_i s_j L q_l \in T,$$

$$q_0 s_b = q_0$$

$$s_b q_0 s_M = q_0 s_M$$

$$s_M q_0 s_M = q$$

Using this semigroup we next construct what is termed, in [11], as Boone's group. Note that each of the relations in $R(T)$ is of the form $F_i q_{i_1} G_i = H_i q_{i_2} K_i$, where i is an element of a finite indexed set I and the F's, G's, H's, and K's are positive s-words or e. If $X = s_{b_1}^{\epsilon_1} s_{b_2}^{\epsilon_2} ... s_{b_m}^{\epsilon_m}$ then denote $X^{\#} = s_{b_1}^{-\epsilon_1} s_{b_2}^{-\epsilon_2} ... s_{b_m}^{-\epsilon_m}$. Then Boone's group, denoted $B(T)$, has the following presentation.

Generators: $q, q_0, ..., q_N, s_0, ..., s_M, x, t, k, r_i, i \in I$; Relations: $\forall i \in I$, and all $b = 0, ..., M$,

$$x s_b = s_b x^2$$

$$r_i s_b = s_b x r_i x$$

$$r_i^{-1} F_i^{\#} q_{i_1} G_i r_i = H_i^{\#} q_{i_2} K_i$$

$$t r_i = r_i t$$

$$t x = x t$$

$$k r_i = r_i k$$

$$k x = x k$$

$$k(q^{-1} t q) = (q^{-1} t q) k.$$

Boone showed that the above has unsolvable word problem and Collins and Miller, in [11], showed that the Boone group has cohomological dimension 2. This allows $B(T)$ to take the place of the group Q in the following construction discussed in [1], [6] and [27].

Let $Q = \langle x_1, ..., x_I \mid R_1, ..., R_K \rangle$ be the presentation for $B(T)$ given above, i.e., $Q = B(T)$. We apologize for the change of notation but the biautomatic group that we wish to construct spans four distinct papers each with different notation. Let

$$\Gamma = \langle x_1, ..., x_I, a_1, ..., a_J \mid x_i^{-1} a_j x_i = W_{ij+},\ x_i a_j x_i^{-1} = W_{ij-},\ R_k = W_k \rangle$$

where the above relations hold for every $1 \leq i \leq I,\ 1 \leq j \leq J,\ 1 \leq k \leq K$ and each $W_{ij\pm}$ and W_k are positive words of length 14 and $2 \mid R_k \mid +8$ respectively in the letters $\{x_1, ..., x_I\}$ and such that no two letter subword appears more than once in the concatenation of all of the $W_{ij\pm}$ and W_k in some order. In [27], it was shown that it is possible to create a word of length J^2 using J distinct letters, such that no two letter subword is repeated, so we need to choose J such that

$$J^2 \geq (2IJ)14 + \sum_{k=1}^{K} (2 \mid R_k \mid +8).$$

Then according to [1], by letting N be the subgroup generated by the a_j's, we have the following properties:

- The short sequence $1 \longrightarrow N \longrightarrow \Gamma \longrightarrow^{p} Q \longrightarrow 1$ is exact, (i.e. $N \triangleleft \Gamma$ and $\Gamma/N \cong Q$;

- N is finitely generated but in general not finitely presentable;
- Γ is torsion free and word hyperbolic, and thus strongly geodesically automatic;
- $\Gamma \times \Gamma$ is short-lex biautomatic ;
- The group P defined by

$$P := \{(\gamma_1, \gamma_2) \mid p(\gamma_1) = p(\gamma_2)\} \subset \Gamma \times \Gamma$$

  is finitely presented, has unsolvable membership problem and unsolvable conjugacy problem.

- P is generated by $\langle (x_i, x_i), (e, a_j), (a_j, e) \mid i \in \{1, ..., I\}, j \in \{1, ..., J\} \rangle$

Because $\Gamma \times \Gamma$ is biautomatic, even though P does not inherit this property, it must inherit the Quadratic time solution to its word problem, so P already has all of the properties that we require except the presence of commuting subgroups. Of course the subgroups $\langle (e, a_j) \mid j \in \{1, ..., J\} \rangle$ and $\langle (a_j, e) \mid j \in \{1, ..., J\} \rangle$ commute but if the algorithm for the braid cryptosystem were applied using these groups in the place of B_L and B_R then hacking the code could be reduced to solving the conjugacy in each coordinate of $\Gamma \times \Gamma$ separately which can be done in O(nlogn) as Γ is word hyperbolic. This would provide no security at all.

Before modifying Γ to produce a group more suited to our application, we first briefly describe the proof in [1] that the conjugacy problem for P is

unsolvable. Note that a word generated by the set $\{(e, x_i)\}$ is an element of P if and only if the corresponding word in just the x_i's is equal to the identity in Q. Therefore as Q has unsolvable word problem, P has unsolvable membership problem. Also noted in [1], in a word hyperbolic group, the centralizer of any element is cyclic. Thus $C_\Gamma(a_j)$ is $\langle a_j \rangle$ and $C_{\Gamma\times\Gamma}((a_j, a_j))$ is $\langle (e, a_j), (a_j, e) \rangle \subset N \times N < P$ Now $N \times N$ is normal in $\Gamma \times \Gamma$, so for any word in $w \in \langle (e, x_i), (x_i, e) \rangle$, we can express $w^{-1}(a_j, a_j)w$ in terms of the generators of $N \times N$, call this word g. We now ask if g is conjugate to (a_j, a_j) in P. Suppose there exists $w' \in P$ such that $(w')^{-1}(a_j, a_j)w' = g$. Then $w'w \in C_{\Gamma\times\Gamma}((a_j, a_j)) \subset P$ and $w' \in P$ so $w \in P$. Therefore we can determine if (a_j, a_j) is conjugate to g in P if and only if we can determine membership in P. But determining membership in P is an unsolvable problem, so P has unsolvable conjugacy problem.

In order for the above proof to apply to the new group we wish to construct, we need two subgroups which play the role of Q and commute with each other, but still act by conjugation on elements in N in manner which is hard to untangle; i.e. in such a way that it is difficult to distinguish between the actions of the two Q's on N. First we choose two distinct Turing machines T and T', each with unsolvable halting problem and form $Q = B(T)$ and $Q' = B(T')$ following the construction above. Next we choose J so that

$$J^2 \geq (2(I + I')J)14 + \sum_{k=1}^{K}(2 \mid R_k \mid +8) \; + \; \sum_{k=1}^{K'}(2 \mid R'_k \mid +8).$$

This allows us to produce positive a_j-words $W_{ij\pm}$, and $W'_{ij\pm}$ each of length

14 and W_k and W'_k each of length $2 \mid R_k \mid +8$, and $2 \mid R'_k \mid +8$ respectively and such that the concatenation of all of these words contains no two letter subword more than once. Next we set

$$G = \langle x_1, ..., x_I, x'_1, ..., x'_{I'}, a_1, ..., a_J \mid x_i^{-1} a_j x_i = W_{ij+},\ x_i a_j x_i^{-1} = W_{ij-},$$

$$R_k = W_k,\ (x'_i)^{-1} a_j x'_i = W'_{ij+},\ x'_i a_j (x'_i)^{-1} = W'_{ij-},\ R'_k = W'_k,\ [x_{i_1}, x'_{i_2}] = 1\rangle.$$

Then the sequence

$$1 \longrightarrow N \longrightarrow G \longrightarrow^{p} Q \times Q' \longrightarrow 1$$

is exact where N is as before and $Q \times Q'$ denotes the internal direct product $\langle x_1, ..., x_I \mid R_k\rangle \times \langle x'_1, ..., x'_{I'} \mid R'_k\rangle$. Finally, we set

$$P = \langle (\gamma_1, \gamma_2) \mid p(\gamma_1) = p(\gamma_2)\rangle \subset G \times G.$$

Note that although we will show that it is automatic, G is not word hyperbolic. From the previous section we know that word hyperbolic groups can not have torsion-free abelian subgroups that are not cyclic and any pair x_{i_1}, x'_{i_2} would generate a torsion-free abelian group of rank two. Define the following subgroups of G,

$$G_X = \langle x_i, a_j \mid 1 \leq i \leq I,\ 1 \leq j \leq J\rangle$$

$$G_{X'} = \langle x'_i, a_j \mid 1 \leq i \leq I',\ 1 \leq j \leq J\rangle$$

It is important to realize that, although we defined Γ and G_X on the same subset of generators G and they do have some relations in common,

the subgroup G_X is not actually isomorphic to Γ because the x_i''s in G will introduce new relations among the a_j's that will be valid in G_X but not in Γ.

For example suppose we have a sequence $\{c_1, ..., c_m\}$ where $c_i = w_i^{-1} R_i^{\epsilon} w_i$, the w_i's are words in the x_i's, $\epsilon = \pm 1$ and $\prod c_i$ is freely equal to e. Since each $R_k^{\epsilon} W_k^{-\epsilon} = e$ in G, we have

$$\prod_{i=1}^{m} w_i^{-1} R_k^{\epsilon} W_k^{-\epsilon} w_i = e.$$

But this identity together with $\prod c_i = e$ implies that

$$\prod_{i=1}^{m} w_i^{-1} W_k^{-\epsilon} w_i = e.$$

Since N is normal in G, we can express $\prod_{i=1}^{m} w_i^{-1} W_k^{-\epsilon} w_i$ as a word in the a_j's alone. Thus this relation would be valid in Γ but not in Γ' (defined below) but it would also be valid in both G_X and G_X'.

This fact will be important for the security of the cryptosystem we design. Thus Γ is word hyperbolic but G_X may not be. We will define Γ' to be the word hyperbolic group

$$\Gamma' = \langle x_1', ..., x_{I'}', a_1, ..., a_J \mid (x_i')^{-1} a_j x_i' = W_{ij+}',\ x_i' a_j (x_i')^{-1} = W_{ij-}',\ R_k' = W_k' \rangle.$$

Again, it will not be isomorphic to $G_{X'}$ although every relation in Γ' will also be valid in $G_{X'}$. Note that some of the extra relations in the a_j's valid in $G_{X'}$, which are not valid in Γ', will be valid in Γ, and similarly some of the relations in the a_j's which are valid in G_X but not in Γ will be valid in Γ'.

A final note on these extra relations is that they are necessarily longer than the shortest means of expressing the same elements as words using x_i's, x_i''s and the a_j's. This is because the construction calls for replacing words of the form $x_i^{-\epsilon}a_jx_i^{\epsilon}$ with its corresponding $W_{ij\epsilon}$ and any two of these concatenated and reduced, provided they are not the inverse of each other, will give a word of length at least $(14-1)+(14-1)=26$ since only the last letter of the first $W_{ij_1\epsilon}$ could possibly cancel with the first letter of the last $W_{ij_2\epsilon}$, since distinct $W_{ij\epsilon}$'s have no two letter subwords in common. Compare this to the length of $x_i^{-\epsilon}a_{j_1}a_{j_2}x_i^{\epsilon}$.

Because the direct product of automatic groups is automatic, in order to ensure that we can reduce an element of $G\times G$ to its normal form in quadratic computation time we need only show that G is automatic. Furthermore if we can show that G is shortlex automatic, for at least one ordering of the generators, then we have the option of using the software package KMAG developed by Derek Holt et. al. to algorithmically determine an explicit presentation of the automatic structure.

Theorem 25 *The group G defined above has a shortlex automatic structure (A, L').*

Naturally A is taken to be the set of generators of G and their inverses. We place them in the following descending order $\{x_1,\ x_1^{-1},\ ...,x_I,\ x_I^{-1},\ x_1',\ (x_1')^{-1},\ ...,\ x_{I'}',\ (x_{I'}')^{-1},\ a_1,\ a_1^{-1},\ ...,\ a_J,\ a_J^{-1}\}$. Then in the shortlex order, $v<w$ if and only if v is shorter than w or they have the same length and v

comes before w in the lexicographical order with respect to the above ordering of the generators. Recall that this gives a well ordering.

Let L_1 and L_2 be regular languages consisting of the shortlex geodesics of Γ and Γ' respectively, according to the descending order $\{x_1, x_1^{-1}, ..., x_I, x_I^{-1}, a_1, a_1^{-1}, ..., a_J, a_J^{-1}\}$, and $\{x_1', (x_1')^{-1}, ..., x_{I'}', (x_{I'}')^{-1}, a_1, a_1^{-1}, ..., a_J, a_J^{-1}\}$, on their generators respectively. We are assured that such languages are indeed regular since word hyperbolic groups are shortlex automatic with respect to any ordering of the generators. Let $X = \{x_1, ..., x_I, x_1^{-1}, ..., x_I^{-1}\}$ and let $X' = \{x_1', ..., x_{I'}', (x_1')^{-1}, ..., (x_{I'}')^{-1}\}$ and let $\mathcal{A} = \{a_1, ..., a_J, a_1^{-1}, ..., a_J^{-1}\}$ as unordered sets. Then clearly $M_1 = X^*$ and $M_2 = (X')^*$ are regular. Letting ϵ denote the empty word we have, by the properties of regular languages discussed in the last section, that the following languages are regular, since they are formed by concatenating and taking the star closure of regular languages.

$$\mathcal{L}_1 = (M_2 \cup \epsilon)(L_1 M_2)^*(L_1 \cup \epsilon)$$

and

$$\mathcal{L}_2 = (M_1 \cup \epsilon)(L_2 M_1)^*(L_2 \cup \epsilon).$$

Next we note that since there are a finite number of x_i's and x_i''s the set $M = \{x_{i_1}^{\pm 1}(x_{i_2}')^{\pm 1}\}$ is finite and therefore regular and thus $X_2 = A^* M A^*$ is also regular. We now make the following claim.

Lemma 6 *The regular set*

$$L = \mathcal{L}_1 \cap \mathcal{L}_2 - X_2$$

contains the set of shortlex geodesics of G, with respect to the above ordering of the generators.

To prove this we first note that since the x_i's commute with the x_i''s, for any $w_1, w_2 \in G$, we have $w_1 x_{i_1}^{\pm 1}(x'_{i_2})^{\pm 1} w_2 = w_1 (x'_{i_2})^{\pm 1} x_{i_1}^{\pm 1} w_2$ in G. However, since the x_i's are greater than the x_i''s in the above ordering, we know that $w_1 x_{i_1}^{\pm 1}(x'_{i_2})^{\pm 1} w_2$ can not be the shortlex presentation of the element because it is larger than $w_1 (x'_{i_2})^{\pm 1} x_{i_1}^{\pm 1} w_2$ according to the shortlex ordering on $(A \cup X \cup X')^*$. Therefore any shortlex geodesic of the Cayley graph of G can not be a word in X_2.

Next we recall that the extra relations among the a_j's which may not be valid in Γ and/or Γ' all have a shorter presentation using the x_i's, x_i''s and the a_j's, and so would not be present as subwords of geodesics of the Cayley graph of G. Thus when reducing a word in A^* to its shortlex geodesic if it contains a subword which using the relations a_j's has a shorter form, then this shorter form can still be reached by using the shorter presentation of the relations in the x_i's, x_i''s and the a_j's. Therefore, when reducing a word in A^* to its shortlex geodesic, we need only use the relations which are valid in Γ and/or Γ'.

Thus if $w \in A^*$ is a shortlex geodesic, then w has expression $w = w_1 v_1' w_2 v_2' ... w_n v_n'$ where each $w_i \in (X \cup A)^*$ and each $v_i' \in (X')^*$ and possibly w_1 or v_n' or both is the empty word. Then each w_i must be a shortlex geodesic of the Cayley graph of Γ or else we could replace it with it's shortlex geodesic and thereby give a new word u such that $u < w$ and $\overline{u} = \overline{w}$, con-

tradicting the assumption that w is a shortlex geodesic. Thus any shortlex geodesic of the Cayley graph of G must be a word in $\mathcal{L}_1$. Similarly it must also be a word in $\mathcal{L}_2$ so it must lie in L. This completes the proof of the lemma.

By theorem 19 of section 5.4, to show G has a shortlex automatic structure, it is sufficient to show that (A, L) is an automatic structure for G. However, we prefer to find $L' = \mathit{shortlex}(G, A)$ explicitly and show that (A, L') is an automatic structure. $\mathit{shortlex}(G, A)$ denotes the set of shortlex geodesics of the Cayley graph of G with respect to the ordered generating set A. We have shown that any word in L' is also in L but in fact they are not equal. This is because a word $w \in L$ written as $w = w_1 v_1' w_2 v_2' ... w_n v_n'$, for example could have a reduction in a subword $u = w_i v_i' w_{i+1}$ even though w_i and w_{i+1} are both geodesics, u has no subwords of the form $x_{i_1}^{\pm 1}(x_{i_2}')^{\pm 1}$, and the largest subword of u contained in $G_{X'}$ is also a geodesic. To illustrate this, let $R_k = W_k$ be one of the defining relations of Γ and let R_{k1}, R_{k2} be front and end subwords respectively, which when concatenated give R_k. That is, $R_k = R_{k1}R_{k2}$ as an equivalence of words in A^* rather than an equivalence of group elements. R_{k2} could be the empty word. Similarly let $W_k = W_{k1}W_{k2}$, chosen so that the word $W_{k1}^{-1}R_{k1}$ is larger than $W_{k2}R_{k2}^{-1}$ in the shortlex ordering. Now, suppose $w_i = w_{i1}w_{i2}$ where w_{i1} is any word in $(X \cup \mathcal{A})^*$ and $w_{i2} = W_{k1}^{-1}$ or $w_{i2} = W_{k1}^{-1}z_1$ where $R_k = z_1 z_2$ in A^*. Further suppose v_i' is any word in $(X')^*$. Finally suppose $w_{i+1} = w_{i+1,1}w_{i+1,2}$ in A^* where $w_{i+1,1} = R_{k1}$ if $w_{i2} = W_{k1}^{-1}$ or $w_{i+1,1} = z_2$ if $w_{i2} = W_{k1}^{-1}z_1$

and $w_{i+1,2}$ is any word in $(X \cup \mathcal{A})^*$. Then in G the following equalities hold: $u = w_{i1}W_{k1}^{-1}R_{k1}v_i'w_{i+1,2} = w_{i1}W_{k2}R_{k2}^{-1}v_i'w_{i+1,2}$ and u is larger than $w_{i1}W_{k2}R_{k2}^{-1}v_i'w_{i+1,2}$ in the shortlex order.

If we treat the relations $x_i^{-\epsilon}a_jx_i^{\epsilon} = W_{ij\epsilon}$ as being of the same form except with $x_i^{-\epsilon}a_jx_i^{\epsilon}$ taking the place of R_k and noting that $w_{i+1,1}$ must always be x_i^{ϵ}, then we get similar possibilities for reductions of words in L which are not in L'. We can also use the inverse relations $R_k^{-1} = W_k^{-1}$ in the same way. Fortunately however, in each of the decompositions the words w_{i1}, $w_{i+1,2}$ and v_i' can be any words in $(X \cup \mathcal{A})^*$, $(X \cup \mathcal{A})^*$, and $X'(X')^*$ respectively, and the remaining subwords w_{i2} and $w_{i+1,1}$ have only a finite number of possibilities. Thus the set of all such subwords of the form of u which have such a reduction form a regular language, namely

$$(X \cup \mathcal{A})^*w_{i2}X'(X')^*w_{i+1,1}(X \cup \mathcal{A})^*$$

where w_{i2} and $w_{i+1,1}$ are taken from a finite set of ordered pairs. Finally we note that reductions of subwords of the form $u = w_i'v_iw_{i+1}'$ are completely analogous and also form a regular language. Therefore let L_3 be the language of all words in A^* which have a subword of the above form. Then

$$L' = L - L_3.$$

Therefore L' is a regular language.

We are now left with the task of showing that (A, L') is an automatic structure for G. To show this we need only show that if $w_1, w_2 \in L'$ such

that $\overline{w_1 b} = \overline{w_2}$ in G for any $b \in A$, then w_1 and w_2 satisfy the k-fellow traveler property also called the Lipschitz property.

AS $\overline{w_1 b} = \overline{w_2}$, and w_2 is a geodesic, by performing a finite number of substitutions using the defining relations of G (and their inverses) we can reduce $w_1 b$ to w_2 via these substitutions.

Let $l(w)$ denote the length of the word $w \in A^*$ and for $g \in G$ let $L_A(g)$ denote the shortest presentation of g according to the shortlex order on A. Now $\overline{w_1 b} = \overline{w_2}$ implies $\overline{w_1} = \overline{w_2 b^{-1}}$ and w_1 is a geodesic so $l(w_1) \leq l(w_2) + 1$. Therefore $l(w_1 b) - l(w_2) \leq 2$.

If we express the defining relations of G as words equal to the identity then, disregarding the relations $[x_{i_1}, x'_{i_2}]$, we can see that the minimum length of the other defining relations is 17. To see this note from the presentation of Q that the minimum length of each R_k is 4 and W_k has length $2 \mid R_k \mid +8$ for a total length of $3(4) + 8 = 20$ and each $W_{ij\pm}$ has length 14 which added to the length $l(x_i^{-\epsilon} a_j x_i^{\epsilon}) = 3$, gives 17. Thus if one of these defining relations is used to make a substitution then the minimum length of the subword replacing the substituted one is 8.

Using the fact that w_1 is a prefix closed geodesic, we must have $w_1 b =_{A^*} yu_1 b$ and the first substitution, if it exists, must be of the form $yu_1 b \rightarrow yu_2$. Recall that no two defining relations of G have a two letter subword in common. Therefore if a substitution occurs of the form $yu_1 z \rightarrow yu_2 z$ then the next substitution, if it exists, can and must use only the leftmost letter of u_2. Furthermore, if b_2 is the first letter of u_2, so that $u_2 = bu_3$ and there is a

subsequent substitution then we must have $y =_{A^*} y_2u_4$ and the substitution must be of the form $y_2u_4bu_3z \rightarrow y_2u_5u_3z$. Hence successive substitutions must 'travel' from right to left along words equivalent to w_1 in G. This implies that if a subword of w_1b is involved in a substitution, then every letter to the right of that subword must also have been substituted.

The language L' is prefix closed so is there is a prefix u of w_1 no part of which is substituted in any of the substitutions which reduce w_1b to w_2 then $\forall t \leq l(u)$, $\overline{w_1^{-1}(t)w_2(t)} = 0$ and $w_1(t)$ and $w_2(t)$ are identical. In the worst case, we can therefore assume that every letter of w_1b is substituted to produce w_2.

We now proceed by induction on the number of substitutions which use the defining relations of G, other than those of the form $[x_{i_1}, x'_{i_2}]$ or $b^{-1}b$ for $b \in A^*$. Note that we treat the substitutions which occur in subwords of the form $w_iv'_iw_{i+1}$ and $w'_iv_iw'_{i+1}$ as single substitutions even though they involve one or possibly several substitutions of relations of the form $[x_{i_1}, x'_{i_2}]$ followed by a single substitution using the defining relations other than $[x_{i_1}, x'_{i_2}]$ and $b^{-1}b$.

By the above argument, $w_1b = u_nu_{n-1}...u_2u_1b$ and w_1b can be reduced to w_2 by the following sequence of substitutions:

$$u_n...u_2u_1b \rightarrow u_n...u_2b_2z_1 \rightarrow u_n...u_3b_3z_2z_1 \rightarrow ... \rightarrow u_nb_nz_{n-1}...z_1 \rightarrow z_n...z_1 =_A w_2.$$

Recall that $l(w_1b) - l(w_2) \leq 2$ and $b_{i+1}z_i =_G u_ib_i$ for $1 \leq i \leq n-1$ and $v_n =_G v_nb_n$, even when the reduction is over a subword of the form $v_iv'_iv_{i+1}$

or $v_i' v_i v_{i+1}'$ where v_i's are in $\{X \cup (A)\}^*$ and v_i''s are in $\{X' \cup (A)\}^*$.

If $t = l(u_n)$, then $(u_n)^{-1} z_n = b_n$, $w_1(t) = u_n$ and $w_2(t) = z_n z_{n-1,1}$ where $z_{n-1,1}$ is either the empty word, the first letter of z_{n-1} or the first and second letter of z_{n-1}. In any case,

$$\overline{w_1(t)^{-1} w_2(t)} = (u_n)^{-1} z_n z_{n-1,1} = b_n z_{n-1,1}$$

so

$$l(\overline{w_1(t)^{-1} w_2(t)}) = l(b_n z_{n-1,1}) \leq 3.$$

In general, if $t = l(u_n ... u_k)$, then $w_1(t) = u_n ... u_k$, and $w_2(t) = z_n ... z_k z_{k-1,1}$ so that

$$\overline{w_1(t)^{-1} w_2(t)} = b_k z_{k-1,1}$$

and

$$l(\overline{w_1(t)^{-1} w_2(t)}) = l(b_k z_{k-1,1}) \leq 3.$$

It remains to show that the distance is bounded when $t = l(u_n ... u_k) + t_2$ where $1 \leq t_2 < l(u_{k-1})$. Then $w_1(t) = u_n ... u_k(u_{k-1}(t_2))$, where $u_{k-1}(t_2)$ is the first t_2 letters of u_{k-1}. Also, $w_2(t) = z_n ... z_k(v_{k-1}(t_2)) v_{k-2,1}$, where $z_{k-1}(t_2)$ is the first t_2 letters of z_{k-1} and $z_{k-2,1}$ is either the empty word, the first or the first and second letters of z_{k-2}. If u_{k-1} is part of a simple reduction involving exclusively the letters of $\{X \cup (A)\}^*$ or $\{X' \cup (A)\}^*$ but not both, then

$$\overline{w_1(t)^{-1} w_2(t)} = (u_{k-1}^{-1}(t_2)) b_k (z_{k-1}(t_2)) z_{k-1,1}.$$

However, $b_k v_{k-1} = u_{k-1} b_{k-1}$ is a relation in G so $(u_{k-1}^{-1}(t_2)) b_k (z_{k-1}(t_2))$ is a subword of the relation and so if m is the maximum length of the defining

relations of G, then

$$l(\overline{w_1(t)^{-1}w_2(t)}) \leq m + 2.$$

Finally we consider the case where $t = l(u_n...u_k) + t_2$ with $1 \leq t_2 < l(u_{k-1})$, and u_{k-1} is of the form $v_{k-1}v'_{k-1}v_{k-1,1}$ or $v'_{k-1}v_{k-1}v'_{k-1,1}$. We deal only with the case where u_{k-1} is of the form $v_{k-1}v'_{k-1}v_{k-1,1}$, the other case being completely analogous. It $t_2 \leq l(v_{k-1})$ then the result reduces to the previous case. The only interesting case occurs when $l(v_{k-1}) < t_2 < l(v_{k-1}v'_{k-1})$. Let $t_3 = t_2 - l(v_{k-1})$. Note that

- $z_{k-1} = s_{k-1}s'_{k-1}$ such that
- $b_k s_{k-1} = v_{k-1}v_{k-1,1}b_{k-1}$ in G and
- v'_{k-1} is identical to s'_{k-1} in A^*.

Thus $\overline{w_1(t)^{-1}w_2(t)} = (v'_{k-1})^{-1}(t_3)v_{k-1}^{-1}b_k(s_{k-1}s'_{k-1})(t_2+\delta)$ where $\delta \in \{0,1,2\}$. If $t_2 \leq l(s_{k-1})$ then $v_{k-1}^{-1}b_k(s_{k-1})(t_2)$ is a subword of a defining relation and so reduces in G to a word of length less than m, and t_3 is bounded by $l(s_{k-1}) < m$ so the $l(\overline{w_1(t)^{-1}w_2(t)}) \leq 2m$. A better bound can be deduced but is unnecessary for the proof. Finally if $t_2 > (s_{k-1})$ then $v_{k-1}^{-1}b_k s_{k-1} = v_{k-1,1}$ which necessarily commutes with v'_{k-1} so $\overline{w_1(t)^{-1}w_2(t)} = (v'_{k-1})^{-1}(t_3)v_{k-1}^{-1}b_k(s_{k-1}s'_{k-1})(t_2+\delta) = (v'_{k-1})^{-1}(t_3)v_{k-1,1}b_{k-1}(s'_{k-1})(t_2+\delta-l(s_{k-1})) = v_{k-1,1}b_{k-1}(v'_{k-1})^{-1}(t_3)(s'_{k-1})(t_2+\delta-l(s_{k-1}))$. Now $l(v_{k-1,1}b_{k-1}) < m$ so we only need a bound for $l((v'_{k-1})^{-1}(t_2-l(v_{k-1})(s'_{k-1})(t_2+\delta-l(s_{k-1})))$. But recall that v'_{k-1} is identical to s'_{k-1} in A^* so the length $l((v'_{k-1})^{-1}(t_2 - l(v_{k-1})(s'_{k-1})(t_2+\delta-l(s_{k-1})))$ is the difference

of the lengths of $(v'_{k-1})^{-1}(t_2 - l(v_{k-1})$ and $(s'_{k-1})(t_2 + \delta - l(s_{k-1}))$. That is $= t_2 - l(v_{k-1}) - (t_2 + \delta - l(s_m(k-1)) = \delta - l(v_{k-1}) + l(s_{k-1}) \leq 2 + m$.

Therefore every pair of words $w_1, w_2 \in L'$ such that $\overline{w_1 b} = \overline{w_2}$, are k-fellow travelers for $k = 2 + 2m$. Therefore (A, L') is a shortlex automatic structure for G.

Before discussing the implementation of the algorithm based on this group, we detail the properties of the group $G \times G$ and its subgroup P which make the cryptosystem possible. First, as the previous proof shows, G and therefore $G \times G$ is shortlex automatic. This means that we can reduce any word in the generators (and their inverses) of $G \times G$ to its unique normal form in quadratic time. This property is inherited by P.

P also has the following two, finitely generated, commuting subgroups. If we take all ordered pairs of words (w_1, w_2) such that $w_1^{-1} w_2 = e$ in Q or $w_1 w_2^{-1} = e$ in Q together with the ordered pairs (x_i, x_i) they generate the first subgroup. The second subgroup is generated by the ordered pairs of similar word equal to the identity in Q' and the pairs (x'_i, x'_i).

Finally P has unsolvable membership and conjugacy problem. We will discuss in the analysis of the algorithm how this should increase security in the cryptosystem.

To implement a variation of the braid cryptosystem in what we term algorithm 1, we also need the subgroups Γ and Γ' which are word hyperbolic and generate words on the same set of generators as subgroups G_X and $G_{X'}$ of G. They should make encoding the algorithm faster because word hyperbolic

groups have a linear time word problem, whereas automatic groups have quadratic time algorithms.

5.6 The algorithm

Now that we have a group with the desired properties, we can construct a cryptographic protocol which utilizes these properties. We actually exhibit two new key exchange algorithms, the first utilizing calculations in word hyperbolic groups to increase the speed of the exchange and the second utilizing the unsolvability of the conjugacy problem for P to greatly increase the security of the key exchange. It should be noted however, that the faster algorithm should still have stronger security than the braid group cryptosystem because solving the conjugacy problem in a generic biautomatic group can currently only be done in exponential time, while the conjugacy problem for the braid group was shown to be solvable in polynomial time. For the faster algorithm (algorithm1), we begin with the assumption that Peter, the programmer of the algorithm, rather than users Alice and Bob, has performed the following tasks.

- Peter chooses two not necessarily distinct Turing machines T and T' each with unsolvable halting problem, constructs G and calculates the automatic structure for $G \times G$.
- Peter generates a finite and hence partial list of relations in the a_j's which hold in G but not in Γ and a finite list of relations in the a_j's

which hold in G but not in Γ.

Normally these tasks wouldn't be included but as they involve a certain amount of choice which affects the speed and security of the algorithm we include them to make the algorithm as general as possible. However, because they only have to be performed once, we do not need to analyze the speed in performing these tasks. The algorithm 1 is as follows.

1. Alice chooses $(a, b) \in N \times N$ and applies different substitutions on each coordinate from a finite and hence partial list of relations in the a_j's which hold in G but not in Γ to produce (a', b'). (Recall $N = \langle a_1, ..., a_J \rangle$. She then chooses words $w_1, w_2 \in X^*$ such that $w_1 = w_2$ in Q, calculates $w_1^{-1} a' w_1$ and $w_2^{-1} b' w_2$ and transmits (a, b) and $c = (w_1^{-1} a' w_1, w_2^{-1} b' w_2)$ to Bob. She keeps (w_1, w_2), and (a', b') secret.

2. Bob applies different substitutions on a and b each from a different finite list of relations in the a_j's which hold in G but not in Γ' to produce (a'', b''). He then chooses $y_1, y_2 \in (X')^*$, such that $y_1 = y_2$ in Q', calculates $y_1^{-1} a'' y_1$ and $y_2^{-1} b'' y_2$ and transmits $d = (y_1^{-1} a'' y_1, y_2^{-1} b'' y_2)$ to Alice. He keeps (y_1, y_2), (a'', b'') secret.

3. Alice computes $(w_1, w_2)^{-1} d (w_1, w_2)$, maps it via a hashing function h to a binary word, and then encodes her binary message m as $m' = h((w_1, w_2)^{-1} d (w_1, w_2)) \oplus m$ to send to Bob. Here $\oplus$ denotes the exclusive or function on binary strings.

4. Bob then computes $h((y_1, y_2)^{-1}c(y_1, y_2))\oplus m'$. Since (w_1, w_2) and (y_1, y_2) commute, $(w_1, w_2)^{-1}d(w_1, w_2) = (y_1, y_2)^{-1}c(y_1, y_2)$ and $h((y_1, y_2)^{-1}c(y_1, y_2))\cdot m' = m$

Recall that the conjugacy problem for $P = \langle (x_i, x_i), (x_i', x_i'), (a_j, 1), (1, a_j)\rangle$ is unsolvable because although we can solve the conjugacy problem in the larger group G, we can't determine to which of $G\backslash P$ or P, the element which performs the conjugation belongs. However, in the above algorithm, $(a, b) \in N \times N \leq P$ and the centralizer of (a, b) is also in $N \times N$ and hence in P and finally $(w_1, w_2) \in P$. If there exists $(z_1, z_2) \in G$ such that $(z_1, z_2)^{-1}(a, b)(z_1, z_2) = (w_1, w_2)^{-1}(a, b)(w_1, w_2)$ then $(z_1, z_2)(w_1, w_2)^{-1}$ centralizes (a, b) so $(z_1, z_2) = c(w_1, w_2)$ for some c in the centralizer of (a, b) in G. Therfore since both c and (w_1, w_2) are in P so must (a_1, z_2) be in P. Therefore in the above algorithm, any solution to the conjugacy problem for (a, b) and $(w_1, w_2)^{-1}(a, b)(w_1, w_2)$ must automatically be in P. However, as we have stated, the conjugacy problem for G should be exponential and once solved we still need to pick (w_1, w_2) out of the coset $C_G((a, b))(w_1, w_2)$.

Suppose, in place of (a, b) in the above algorithm, we took a more generic term $(v_1, v_2) \in P$. Note that $(v_1, e)) \in C_G((v_1, v_2))$ but $(v_1, e) \notin P$ unless $v_1 = e$ in Q. The unsolvability of the word problem of Q makes this impossible to check in general. Therefore, using (v_1, v_2) in place of (a, b) in the above algorithm, we do ensure an unsolvable conjugacy problem. That is, if we find $(z_1, z_2) \in G$ such that $(z_1, z_2)^{-1}(v_1, v_2)(z_1, z_2) = (w_1, w_2)^{-1}(v_1, v_2)(w_1, w_2)$ then there is no algorithm to determine whether or not $(z_1, z_2) \in P$ let

above if $(z_1, z_2) = (w_1, w_2)$. Of course, in order for the secret to be uncovered a hacker only needs a (z_1, z_2) such that $(z_1, z_2)^{-1}(v_1, v_2)(z_1, z_2) = (w_1, w_2)^{-1}(v_1, v_2)(w_1, w_2)$ and such that (z_1, z_2) commutes with the x_j''s. If however, the initial solution to the conjugacy problem in $G \times G$ doesn't yield such a candidate, the deriving one that does from (z_1, z_2) is equivalent to the membership problem of the centralizer of (v_1, v_2) which has been shown to also be unsolvable. Furthermore, (v_1, v_2) can easily be chosen so that the probability that the centralizer of $C_G((v_1, v_2))$ intersects non-trivially with the centralizer of the x_j's is low. This requires the hacker to find the exact (w_1, w_2).

Therefore for even more security we incorporate this technique into the algorithm below. However, this added security may come at a price. Because (v_1, v_2) now contains both x_i's and x_i''s we can not do the calculation that reduces $(w_1, w_2)^{-1}(v_1, v_2)(w_1, w_2)$ to its normal form in a word hyperbolic group; we must instead do all calculation in the automatic group $G \times G$.

For the algorithm with even stronger security, (algorithm 2), we also begin with Peter.

- Peter chooses two not necessarily distinct Turing machines T and T' each with unsolvable halting problem, constructs G and calculates the automatic structure for $G \times G$.

Once this task is performed we move to the actual key exchange.

1. Alice chooses $(v_1, v_2) \in P$ She then chooses words $w_1, w_2 \in X^*$ such that $w_1 = w_2$ in Q, calculates the normal form of $(w_1^{-1}v_1w_1, w_2^{-1}v_1w_2)$, in $G \times G$, and transmits (v_1, v_2) and $c = (w_1^{-1}v_1w_1, w_2^{-1}v_1w_2)$ to Bob. She keeps (w_1, w_2) secret.

2. Bob chooses $y_1, y_2 \in (X')^*$, such that $y_1 = y_2$ in Q', calculates the normal form of $(y_1^{-1}v_1y_1, y_2^{-1}v_2y_2)$ in $G \times G$ and transmits $d = (y_1^{-1}v_1y_1, y_2^{-1}v_2y_2)$ to Alice. He keeps (y_1, y_2) secret.

3. Alice computes $(w_1, w_2)^{-1}d(w_1, w_2)$, maps it via a hashing function h to a binary word, and then encodes her binary message m as $m' = h((w_1, w_2)^{-1}d(w_1, w_2)) \oplus m$ to send to Bob. Here $\oplus$ denotes the exclusive or function on binary strings.

4. Bob then computes $h((y_1, y_2)^{-1}c(y_1, y_2)) \oplus m'$. Since (w_1, w_2) and (y_1, y_2) commute, $(w_1, w_2)^{-1}d(w_1, w_2) = (y_1, y_2)^{-1}c(y_1, y_2)$ and $h((y_1, y_2)^{-1}c(y_1, y_2)) \cdot m' = m$

5.7 Analysis of the Algorithm

For several reasons, at first glance the new cryptosystem might appear to be slower than the braid cryptosystem even though it should actually be faster to implement. First the notation is more complex because we are working at times in a direct product of groups and at other time its coordinate groups. We can consider the direct product an internal one when determining the

length of the elements chosen so that, for example, the element $a \in B_L$ of the braid cryptosystem can have the same length as the sum of the lengths of a and b for $(a, b) \in N \times N$. Also, while the calculations of $(y_1, y_2)^{-1}c(y_1, y_2)$ and $(w_1, w_2)^{-1}d(w_1, w_2)$ will be performed in quadratic time as are their counterparts in the braid cryptosystem, the initial keys $w_1^{-1}a'w_1$, $w_2^{-1}b'w_2$, $y_1^{-1}a''y_1$, and $y_2^{-1}b''y_2$ can be calculated in linear time because they are elements of hyperbolic groups whose elements can be reduced to a normal form in linear time. Thus key exchange in the new cryptosystem should be faster.

Another reason the new cryptosystem looks slower is the need to compute a' from a and b' from b. In practice however, the can be created quickly and simultaneously as follows. Express each of the relations in N which hold in G but not in Γ in the form $r_i = s_i$. Then add the r_i's to the generating set for a and b and add the s_i's to the generating set for a' and b'. We can create w_1 ,w_2 and y_1, y_2 in a similar way. Creating a'' and b'' should be possible in a constant time since it should only involve a few substitutions. The necessity of computing a', a'', b' and b'' results from the fact that Γ and Γ' are word hyperbolic and thus allow for very fast calculations. We wish to exploit this fact when we create and exchange keys, but we want to prevent potential hackers from utilizing this property in trying to recover w_i and y_i from $w_1^{-1}a'w_1$, $w_2^{-1}b'w_2$, $y_1^{-1}a''y_1$, and $y_2^{-1}b''y_2$. Thus since, for example $a' \neq a$ in Γ, $x^{-1}a'x$ is not conjugate to a in Γ. The conjugacy problem can be solved in *nlogn* time in word hyperbolic groups which is too fast to be allowed. The hacker has no choice but to attempt to solve the conjugacy problem in the

biautomatic group, which currently takes exponential time.

It should be noted however that even finding $g_1, g_2 \in \Gamma$ such that $g_1^{-1}ag_1 = w_1^{-1}a'w_1$ and $g_2^{-1}bg_2 = w_2^{-1}b'w_2$ does not give away the secret (w_1, w_2) because there is no way of determining if $g_1 = g_2$ in Q and thus if (g_1, g_2) is (w_1, w_2). However, if the hacker has the original message and the encrypted message he/she can try the (g_1, g_2) and see if he/she recovers the original from the encrypted so it will be necessary from computer trials to determine how often solving the conjugacy problem in $G \times G$ will yield the appropriate solution in P.

As a final note we mention that the cryptosystem may be vulnerable to a length based attack which consists of conjugating $w_1^{-1}a'w_1$ by each element of X in turn to see if the length of $w_1^{-1}a'w_1$ decreases. It may be possible to prevent such an attack by ensuring that the suffix of w_1^{-1} and the prefix of a' form a subword of a defining relation which is more than half the length of said defining relation.

Chapter 6

Locally Finite-Indicable Groups

In 1940, in [13], Graham Higman introduced the term indicable to describe a group having a homomorphism onto a non-trivial subgroup of the additive rationals, i.e. an infinite cyclic factor group. In [6], Burns and Hale proved that locally indicable groups are right orderable. Kopytov and Medvedev explore locally indicable groups in 7.4 of [18] and in particular give a short proof using ultrafilters as introduced by Malcev in [21] that the class of locally indicable groups forms a quasi-variety. Brodskii [7] then proved that if $\mathfrak{R}$ is any quasi-variety then the class L($\mathfrak{R}$), of locally $\mathfrak{R}$-indicable groups– groups having a non-trivial homomorphism onto a $\mathfrak{R}$-group– is equal to the class N($\mathfrak{R}$) of groups which have a normal system with factors in $\mathfrak{R}$. Locally $\mathfrak{R}$-indicable is also called locally $\mathfrak{R}$-projectable and locally $\mathfrak{R}$-decomposable.

The equality $L(\mathfrak{R}) = N(\mathfrak{R})$ for quasi-varieties seems to depend largely on the fact that quasi-varieties are closed under ultra products as defined in [18]

and [21]. To see that closure under ultraproducts is a defining characteristic of quasi-varieties one need only consult 11.1.2 of [21]. It is therefore possible that $L(\mathfrak{R}) = N(\mathfrak{R})$ if and only if $\mathfrak{R}$ is a quasi-variety, but the question remains open. Even for the class $\mathfrak{F}$ of finite groups is merely the union of an ascending sequence of quasi-varieties, but not a quasi-variety itself, $L(\mathfrak{F}) \neq N(\mathfrak{R})$ as we show here.

In section 2, after some preliminary discussion, we exhibit a subclass of $L(\mathfrak{F})$ that is not in $V(\mathfrak{F})$. Then, in section 3, we embed each member of the subclass in its own 2-generator group to create a class of finitely generated groups which are locally finite indicable but do not have normal systems with finite factors.

6.1 Groups Without Normal in $\mathfrak{F}$

For arbitrary class $\mathfrak{R}$, the inequality $L(\mathfrak{R}) \supseteq N(\mathfrak{R})$ is easily proved. If $G \in N(\mathfrak{R})$, there exists a totally ordered set Λ such that $\forall \lambda \in \Lambda, \exists D_\lambda, C_\lambda \subseteq G$ which satisfies the following:

1. $C_\lambda \lhd D_\lambda$
2. $D_\lambda / C_\lambda \in \mathfrak{R}$
3. $\mu \subseteq \lambda \Rightarrow D_\mu \subseteq C_\lambda$
4. $\bigcup_{\lambda \in \Lambda} D_\lambda \backslash C_\lambda = G \backslash 1$

Let H be a finitely generated subgroup of G. By properties 3 and 4, there exists λ such that $H \leq D_\lambda$ and $H \cap C_\lambda \neq H$. Therefore

$$1 \neq H/H \cap C_\lambda \simeq HC_\lambda/C_\lambda < D_\lambda/C_\lambda \in \Re,$$

which completes the proof that $L(\Re) \supseteq N(\Re)$.

We therefore restrict our discussion to the reverse inequality from now on. Letting $\Im_n$ denote the class of groups of order at most n, we see that $\Im = \{\Im_n\}_{n=1}^{\infty}$. $\Im_n$ is a quasi variety for each positive integer n, so, if G is a group and there exists a positive integer n such that every finitely generated subgroup of G has a non-trivial homomorphic image of order at most n then G has a normal system with factors of order at most n. In order to show $L(\Im) \neq N(\Im)$, it is therefore necessary to exhibit a group G with the property that every finitely generated subgroup of G has a non-trivial, finite, homomorphic image. But G must also have the property that the orders of these factor groups increase without bound. It is further required that these factors be simple or we could take smaller ones. The simplest such example is the direct limit A_∞ of finite alternating groups A_n, n=1,2,3,... Every finitely generated subgroup of A_∞ is contained in some A_n and so is finite. Hence $A_\infty \in L(\Im)$. However, A_∞ is simple and we now show that it can not have a normal system of finite factors.

By way of contradiction, assume there exists a totally ordered set Λ such that $\forall \lambda \in \Lambda, \exists D_\lambda, C_\lambda \subseteq A_{infty}$ which satisfies the four criteria above with 2

replaced by

$$D_\lambda / C_\lambda is finite.$$

Take any $n > 4$. Then A_n is simple. There exists γ such that$\forall \alpha > \gamma$, $A_5 \leq D_\alpha$ so take λ to be minimal in this respect. Then $\exists m > 5$, such that $A_m \not\leq D_\lambda$ and so $\exists \mu > \lambda$ such that $A_m \leq D_\mu$. Now $C_\mu \cap A_m \lhd D_\mu \cap A_m = A_m$ so A_m simple implies that $C_\mu \cap A_m = e$. But property 3 of the normal systems implies that $D_\lambda \subseteq C_\lambda$ so $C_\mu \cap A_m \supseteq A_n$ giving a contradiction.

A_∞ is one group lying in $L(\Im)\backslash N(\Im)$, but is by no means an exceptional example. In fact combining theorems 4.5 and 4.6 of [16], we get

Theorem 1 *if G is a countable infinite, locally finite, simple group isomorphic to a subgroup of $GL_n(F)$, for some field F and positive integer n, then G is the union of an ascending sequence of finite subgroups almost all of which are simple.*

Groups meeting the criteria above must also lie in $L(\Im)\backslash N(\Im)$.

6.2 Two-generator Groups

Perhaps the only inelegance of the members of $L(\Im)\backslash N(\Im)$ above is that, like A_∞ for example, they can not be finitely generated. However [23] gives a method for embedding any countable group G two-subnormally in a two-generator group H. After presenting the embedding, we show that it preserves locally finite indicability. Furthermore, since the embedding is sub-

normal, the resulting group H can not have a normal system of finite factors, given that G doesn't.

It should be noted that [23] deals strictly with ordered groups so the embedding is constructed so that the order on G can be extended to an order on H. Although this makes the construction slightly more cumbersome than necessary for our purposes, it in no way restricts us to using ordered groups.

6.2.1 Embedding in a Two-generator Group

Following [23], if G is any countable group we begin with the sequence of subgroups of CrG^Z:

$$G^{(0)} = \langle x \mid x_i = e \; if \; i \neq 0, \; x_0 = g, \; g \in G \rangle$$

and for $k \geq 0$,

$$G^{(k+1)} = \langle x \mid x_i = e \; if \; i < 0, \; x_i = g^{\binom{k+i}{k}} \; if \; i \geq 0, \; g \in G \rangle,$$

Let t be the automorphism of CrG^Z defined by $G_i^t = G_{i+1}$ and

$$K = \langle \; t, \; G^{(i)} \; \mid \; i \geq 0 \; \rangle$$

Note that $[t, G^{(k+1)}] = G^{(k)}$ so the $G^{(i)}$ generate K' with $G^{(0)}$ normal in K'. The countability of G implies that K is also countable and so can be well ordered as $K = \{a_1, a_2, a_3, ...\}$. Let u be the element of CrK^Z defined by:

$$u_{2^m} = a_m$$

$u_n = e \; if \; n \neq 2^m, \; m = 1, 2, ...$

Finally, we let τ be the automorphism of CrK^Z defined by $K_i^\tau = K_{i+1}$. Our desired H is $\langle u, \tau \rangle$.

6.2.2 Two-generator Groups in $L(\Im)\backslash N(\Im)$

We are left with the task of proving the following theorem:

Theorem 2 *If G is a countable group in $L(\Im)\backslash N(\Im)$ and H is the two-generator group constructed from G as above, then H also lies in $L(\Im)\backslash N(\Im)$.*

Proof As mentioned earlier, if G does not have a normal system of finite factors then neither can H. Assume every finitely generated subgroup of G has a non-trivial homomorphism onto a finite group. Then we show as an intermediate step that K in the above construction also has this property. Let L be a non-trivial, finitely generated subgroup of K. $L' = L \cap CrG^Z$ just as $K' = K \cap CrG^Z$ so if $L = L'$ then $L < CrG^Z$. At least one of the i^{th} components of one of the generators of L are non-trivial, so the projection of the generators of L onto this i^{th} component yields a homomorphism from L onto a finitely generated subgroup of G, which, by assumption, has a finite factor. Otherwise, if $L \neq L'$, then L/L' is a nontrivial, finitely generated abelian group and so also has a non-trivial finite factor. Thus if G is in $L(\Im)\backslash N(\Im)$, then so is K. $H \in L(\Im)\backslash N(\Im)$ is proved in similar manner – a finitely generated subgroup J of H is either a subgroup of CrK^Z or J/J' is a finitely generated abelian group. □

Bibliography

[1] Baumslag, G.; Bridson, M.R.; Miller, C.F.; Short, H., "Fibre products, non-positive curvature, and decision problems," Comment. Math. Helv. 75(2000) 457-477.

[2] Baumslag, G.; Gersten, S.M., Shapiro, M.; Short, "Automatic Groups and Amalgams" *Algorithms and Classification in Combinatorial Group Theory*, Springer-Verlag New York, Inc. (1992) 179-194.

[3] Bokut', L.A., "Malcev's problem and groups with a normal form," *Word Problems II*, North-Holland Publishing Company (1980) 29-53.

[4] Bokut', L.A., "On the Novikov groups," Algebra i Logika, Seminar, 6 (1967), no. 1, 25-38.

[5] Bokut', L.A.,"Degrees of unsolvability of the conjugacy problem for finitely-presented groups", Algebra i Logika, 7(1968), no. 6, 4-52.

[6] Bridson, M.R.; Haefliger, A., *Metric Spaces of Non-positive Curvature*, Grundlehren der Mathematischen Wiss., Vol. 319, Springer-Verlag, Heidelberg (1999).

[7] Brodskii, S.D., "Equations over groups and groups with only one defining relation," Sibirsk. Mat. Zh, 25(1989), no.2, 84-103.

[8] Burns, R.G.; Hale, V.W.D, "A note on group rings of certain torsion-free groups," Canad. Math. Bull. 15(1972), 441-445.

[9] Cannon, J.W.; Epstein, D.B.A.; Holt, D.F.; Levy, S.V.F.; Patterson, M.S.; Thurston, W.P., *Word Processing in Groups*, Jones and Bartlett Publishers, Inc., Boston, 1992. xi+330pp.

[10] Cheon, J.; Jun, B., "A polynomial time algorithm for the braid Diffie-Hellman conjugacy problem," Proc. of Crypto 2003, Lecture Notes in Computer Science, 2729 (2003), pp. 212-225.

[11] Collins, D.J.; Miller III C.F., "The word problem in groups of cohomological dimension 2". *Groups St. Andrews in Bath*, LMS Lecture Notes 270 (1998), pp. 211-218.

[12] Glass, A.M.W.; Holland, C. (Ed.) *Lattice-Ordered Groups. Advances and Techniques*, Mathematics and its Applications, 48. Kluwer Academic Publishers Group, Dordrecht, 1989.

[13] Higman, G."The units of group rings," Proc. London Math. Soc., 46 (1940), no.2, 231-248.

[14] Higman, G.; Neumann, B.H.; Neumann, H., "Embedding theorems for groups" J. London Math. Soc. 24 (1949)247-254.

[15] Ko, K.H.; Lee, S.J.; Cheon, J.H.; Han, J.W.; Kang, J.; Park, C., "New public-key cryptosystem using braid groups," website http://www.tcs.hut.fi/ helger/crypto/link/public/braid/

[16] Kegel, O.H.; Wehrfritz, B.A.F., *Locally Finite Groups*. North-Holland Mathematical Library, vol. 3, North-Holland Publishing Co., Amsterdam-London; American Elsvier Publishing Co., Inc., New York, 1973. xi+210pp.

[17] Khukhro, E., *Nilpotent Groups and Their Automorphisms*, Walter de Gruyter & Co., Berlin, 1993. xiii+252pp.

[18] Kopytov, V.; Medvedev, N. *Right Ordered Groups*, Siberian School of Algebra and Logic, New York 1996. ix+250pp.

[19] Kopytov, V.M.; Medvedev, N. Ya. *The Theory of Lattice-Ordered Groups*, Kluwer Academic Publishers, The Netherlands, 1994. xiii+400pp.

[20] Lee, E.; Park, J.H., "Cryptanalysis of the public-key encryption based on braid groups," website http://www.tcs.hut.fi/ helger/crypto/link/public/braid/.

[21] Malcev, A.I. *Algebraic Systems.* Springer-Verlag, New York-Heidelberg, 1973. xii+317pp.

[22] Magnus, Wilhelm; Karrass, Abraham; Solitar, Donald *Combinatorial Group Theory*, Interscience Publishers, New York (1966).

[23] Mura, R.B.; Rhemtulla, A. *Notes on Orderable Groups*, University of Alberta, Edmonton (1975).

[24] Novikov, P.S., "Unsolvability of the conjugacy problem in group theory," Izv. Akad. Nauk. SSSR, Ser. Matem. 18 (1954), no. 6, 485-524.

[25] Olshanskii, A. Yu.; Sapir, M.V., "Length and Area Functions on Groups and Quasi-Isometric Higman Embeddings," International Journal of Algebra and Computation, 11 (2001), No. 2, 137-170

[26] Passman, D.S. *The Algebraic Structure of Group Rings*, John Wiley & Sons, Inc., New York, 1977. xiv+720pp.

[27] Wise, D., "Incoherent negatively curved groups," Proc. Amer. Math. Soc. 126(1998), 957-964.